解　説

統計手法を導入した文法的パターン認識法†

周 藤 安 造††

1. は じ め に

パターン認識には，多くの手法があるが，これを大きく分類すると統計的な方法と文法的な方法となる[1)~3)]．このうち，後者はいわゆる Syntactic または Linguistic approach と呼ばれている方法であり，図形言語的パターン認識法とか構造解析的パターン認識法などと言うこともある．本稿では，これを便宜上**文法的パターン認識法**と呼ぶことにする．

2つのパターン認識法には，パターンを数理統計的に解析し，認識するか，文法構造的に解析し，認識するかという基本的な違いがある．統計的パターン認識法は，パターンの大きさ，長さ，面積，角度，形状などの各種情報量を特徴パラメータとして抽出し，これらのパラメータ集合（ベクトル値）によって，もとの冗長なパターンを代表記述する方法であり，パターンの分類は，普通パラメータ集合で構成される特徴空間をパターン・クラスごとに最適空間分割することによって行われる[4),5)]．

このような統計的パターン認識法は過去 10 年間においてめざましく発達し，文字認識，医用画像の診断，リモート・センシングにおける作物の認識などにおいて，技術の実用化が進んでいる[4)]．

ところで，パターン認識問題においては，パターンを構成する情報の構造的な解析が重要となる分野が少なからず存在しており，たとえば図形の形状認識[6)~10)]，一次元波形解析[6),11),12)]，風景，物体の解析[6),13)]などがこれに相当する．このような問題を統計的パターン認識法によって解決しようとすると，一般に扱うべき特徴パラメータの個数が膨大となってしまい，実用的でなくなることから**文法的パターン認識法**の有効性が見い出された．

文法的パターン認識法は，パターンのもつ情報構造を文法的な側面から解析し，これを文章または言語構造と見なして，その性質を認識する方法であり一般に統計的パターン認識法に比べきめの細かい情報構造の認識が可能とされている．さてこの方法では，まずパターンを構成する基本的な要素をパターン・プリミティブまたは単にプリミティブとして定義するが，（図形の場合，線分，角度，曲率など）これは文章にたとえればひとつひとつの単語に相当する．続いてパターンの情報構造を記述するための文法，すなわちパターン生成文法をパターン・プリミティブ群および文法書き換え規則などから構成する．この方法でのパターンの認識は，普通オートマトンを適用して行うが，その場合，もとのパターンをプリミティブ単位に解析し，その結果パターンがあらかじめ定義した文法どおりの情報構造になっていれば，正しく認識されたことになり，さもなければ認識されない．

文法的パターン認識法においては，パターン・プリミティブをどのように選ぶか，またパターン・プリミティブをベースにどのような文法を構成するかといったことが認識方式の良し悪し，認識精度，処理効率などに重要な影響を与えるが，このような**文法的パターン認識法**の概要は**図-1**のようになる．

この図において，まずサンプル・パターンから妥当なパターン・プリミティブを選択し，もとのパターンをプリミティブ列に分解することになる．

次に，各パターン・クラスのプリミティブ列を記述するための文法 $G_1, G_2, \cdots, G_n$ を構成し，さらにこれらを合成した文法Gを全サンプル・パターンに対する文法として求める．

一方，テスト・パターンに対する処理は，サンプル・パターンについて選択したパターン・プリミティブの抽出から始まる．これは一般的なパターン認識機械の Receptor の部分に相当しており，これによって，特徴パラメータに相当するプリミティブ列が出力される．次の構文解析 (Syntax analyzer) は文法Gにもとづいて構成されるが，これは，いわゆる Categorizer に相当しており，有限オートマトンなどによってテスト・

† Syntactic Pattern Recognition Complemented by Stochastic Techniques by Yasuzo SUTO (Medical Engineering Laboratory, Toshiba Corporation).

†† 東京芝浦電気(株)医用機器技術研究所

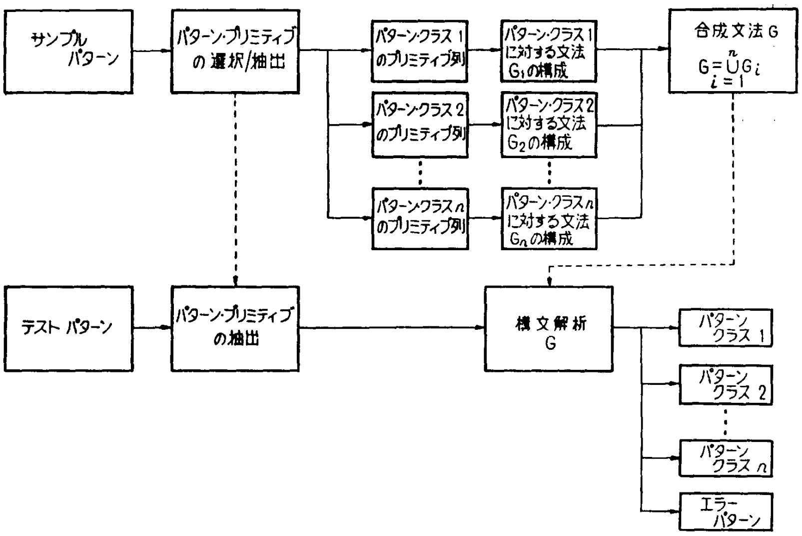


図-1 文法的パターン認識の処理過程

パターンが n 種類のクラスに分類される．

このようなパターン認識法において，パターン・プリミティブの計算機による自動的な抽出（セグメンテーション）は必ずしも容易でなく，もとのパターンに存在するノイズとかパターンの“ばらつき”にもとづく変形などにより，セグメンテーション・エラーとかプリミティブの誤抽出が起り，不正確なパターン記述の原因となる．

そこで，近年不正確な条件下でノイズ・パターンあるいは変形パターンを記述する方法がいろいろ試みられている．すなわち，図-1で示す文法は，プロトタイプの文法によってのみ記述できる標準的なパターンを扱うが，ノイズ・パターンとか変形パターンなどのいわゆるエラー・パターンについても，文法規則に統計的な概念を導入し，パターン認識アルゴリズムを同定していこうとする判別文法 (Discriminant Grammar) あるいは**エラー修正文法** (Error Correcting Parsing) が提案されている[2),3)]．

本稿では，このような考え方にもとづいて**文法的パターン認識法**に統計手法をとり入れ，認識処理に柔軟性と効率化をもたらす方法を示すが，その前に**文法的パターン認識法**の概要について述べる．

文法的パターン認識法の概要では，一次元文法と高次元文法について紹介している．

2. 文法的パターン認識法の概要

パターンを一次元記号列（ストリング），木構造（ツリー）などによって記述するアルゴリズムをパターン文法というが，もとの一次元記号列，木構造などがどのような構造になっているかにより，具体的な文法構成法が異なる[14)~29)]．すなわち，一次元記号列に対しては，もとのパターンの性質により，正規文法，文脈自由文法，文脈依存文法などが構成される．また，より複雑なパターンの記述に対しては，高次元文法が適しており，このような文法として木構造文法[20)~25)]，ウェッブ文法[27)]，グラフ文法[28)]，ネットワーク文法[29)]などが構成される．

一方，パターンの構文解析（認識）はオートマトンによって行われるが，オートマトンは文法に対応して構成される．すなわち，正規文法に対して有限オートマトン，文脈自由文法に対して線形有界オートマトン，木構造文法に対して木構造オートマトン，ウェッブ文法に対してウェッブ・オートマトンがそれぞれ構成される．

文法的パターン認識法においては，このようなオートマトンによってパターンが正しく構文解析できれば（文法にそって正しく記述されているなら），目的とするパターンが認識されたことになり，さもなければ認識できなかったことになる．

以下において，一次元文法，高次元文法によるパターンの記述と認識法について述べる．

2.1 一次元文法による構造記述と認識

パターンの構造記述は一次元言語を対象としたチョムスキーの句構造文法にその起源がある．一般に句構

造文法は次の4字組で与えられる．

$$\mathbf{G}=(V_N, V_T, R, S) \quad (1)$$

ここで

① V_N は語いと呼ばれる非終端記号からなる有限集合

② V_T は終端記号からなる有限集合

③ R は順序対 (α, β) の有限集合であり，次の形式で与えられる

$$\alpha \to \beta, \alpha, \beta \in V^* \ (V=V_N \cup V_T)$$
$$\alpha \neq \lambda \ (\text{空集合})$$

④ S は始端記号で V_N に属す

いま，任意のパターン $x \Cap V_T^*$ が文法 **G** によって生成可能なとき，これを記法

$$S \overset{G}{\Rightarrow} x$$

によって与えることにすると，文法 **G** によって定義されるパターン全体は

$$L(\mathbf{G})=\{x \mid x \in V_T^*, S \overset{G}{\Rightarrow} x\} \quad (2)$$

となる．

このような文法は一般に0型文法と呼ばれるが，これに制約を与えることによって，いくつかの文法のクラス分けがなされる．すなわち，文法 **G** の書き換え規則 R: $\alpha \to \beta$ において，つねに $|\beta| \geqq |\alpha|$ なら文法 **G** は文脈依存型であり，$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2 (\alpha_1, \alpha_2, \beta \in V^*, \beta \neq \lambda, A \in V)$ のように A が α_1 と α_2 の間に現れたときだけ A が β に置き換えられる．なお，$|\beta|$ は β の一次元記号列を構成する要素の数を表す．ここで，文法 **G** に対し，さらに $|\alpha|=1$ という制約が加わると，これは文脈自由文法となり，α は1個の変数で，それがどのような文脈に現れるか無関係に β と置き換えられる．また，$A \to aB, A \to a (A, B \in V_N, a \in V_T)$ のような形式の書き換え規則からなる文法を正規文法という．

正規文法によって記述されたパターンを認識する機械，すなわち有限オートマトン M は次の形式で与えられる[30]．

$$M=(K, \Sigma, \delta, q_0, F) \quad (3)$$

いま，x を一次元記号列からなる入力パターンとし $(x \in \Sigma^*)$, $\delta(q_0, x)=p$ なる p が F に属するとき，パターン x は有限オートマトン M によって受理(認識)されたことになる．有限オートマトン M によって受理(認識)されるすべてのパターン x の集合を T(M) で与えると

$$T(M)=\{x \mid \delta(q_0, x) \in F\} \quad (4)$$

と表せる．

一般に，L(G)=T(M) なら，文法 G はオートマトン M に等しいという．このような関係を正規文法 G と有限オートマトン M について例を示す[30]．

いま，正規文法を

$$\mathbf{G}=(V_N, V_T, R, S)$$

で与える．

ここで，書き換え規則 R を

$$S \to 1A, \ S \to 0B, \ A \to 0C, \ B \to 1C$$
$$C \to 0A, \ C \to 1B, \ A \to 1, \ B \to 0$$

で与え，

$$V_N=\{S, A, B, C\}$$
$$V_T=\{0, 1\}$$

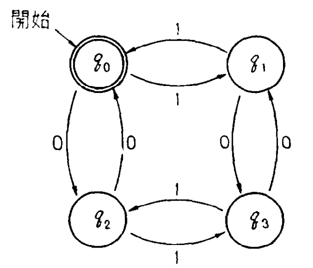
とすると，文法 **G** は偶数個の0と1からなる一次元記号列の集合を記述する．

このような正規文法 **G** に対する有限オートマトン M および M の状態遷移図を**図-2**に示すが，M によって，$\delta(q_0, x)=q_0$ となるような一次元記号例 x のみが受理(認識)されることになる．(ただし，$x \in \Sigma^*$)

さてここで染色体を文脈自由文法によって記述した例を**図-3**に示す[31]．3種類の染色体 Median, Submedian, Acrocentric およびそれらを記述する文法 $\mathbf{G_M}$, $\mathbf{G_S}$, $\mathbf{G_A}$ がそれぞれ与えられている．

2.2 高次元文法による構造記述と認識

パターンのもつより複雑な情報構造を記述するには，一般に高次元文法の方が適している．このような高次元文法には，木構造文法，ウェッブ文法，グラフ文法などがある．このうち，K. S. Fu らによる木構造文法の構成法，パターン記述法などについて紹介す



$M=(K, \Sigma, \delta, q_0, F)$
$\Sigma=\{0, 1\}$
$K=\{q_0, q_1, q_2, q_3\}$
$F=\{q_0\}$
$\delta(q_0, 0)=q_2, \ \delta(q_0, 1)=q_1$
$\delta(q_1, 0)=q_3, \ \delta(q_1, 1)=q_0$
$\delta(q_2, 0)=q_0, \ \delta(q_2, 1)=q_3$
$\delta(q_3, 0)=q_1, \ \delta(q_3, 1)=q_2$

図-2 オートマトン M の状態遷移図

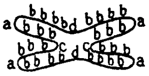
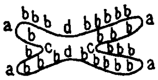
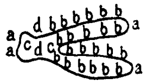
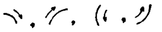
〔Median〕	〔Submedian〕	〔Acrocentric〕
		
cbbbabbbbdbbbbabbb cbbbabbbbdbbbbabbb	cbabbbdbbbbbabbb cbbbabbbbbdbbbab	cadbbbbbbabbbbb cbbbbbabbbbbbda
$G_M=(V_{NM}, V_{TM}, R_M, S)$ $V_{NM}=\{S, A, B, D, H, J, E, F\}$ $V_{TM}=\{a, b, c, d\}$ (ここで，a, b, c, d はそれぞれプリミティブ  を与える) R_M: S→AA D→FDE E→b A→cB D→d H→a B→FBE F→b J→a B→HDJ	$G_S=(V_{NS}, V_{TS}, R_S, S)$ $V_{NS}=\{S, A, B, D, H, J, E, F, W, G, R, L, M, N\}$ $V_{TS}=V_{TM}$ R_S: S→AA D→FDE G→FG J→a A→CM D→FG W→WE L→HNJ B→FBE D→WE F→b R→HNJ B→FL L→FL E→b G→d B→RE R→RE H→a W→d M→FBE N→FDE	$G_A=(V_{NA}, V_{TA}, R_A, S)$ $V_{NA}=\{A, B, D, H, J, E, F, L, R, W, G\}$ $V_{TA}=V_{TM}$ R_A: S→AA D→FG G→FG R→HDJ E→b A→cB D→WE W→WE G→d F→b B→FL L→FL L→HDJ W→d B→RE R→RE H→a J→a

図-3 染色体のコーディングと文法

る[20]．

木構造文法 G_t は次の4字組で与えられる．

$$G_t=(V, r, P, S) \tag{5}$$

(5)式において，$V=V_N \cup V_T$, V_N は非終端記号，V_T は終端記号．S は始端記号で，$S \in V_N$．P は非終端記号からサブ木構造を生成する書き換え規則であり，一般には次のような形式をとる．

$$P:\ X_0 \to \begin{array}{c} x \\ / \ \backslash \\ X_1 X_2 \cdots X_{r(x)} \end{array} \qquad X_0 \to x$$

ここで，$x \in V_T$, $X_0, X_1, \cdots, X_{r(x)} \in X_N$ で，r は特定な節（ノード）x から枝分かれしている非終端または終端記号の個数を与える次のような関数である．

$$r:\ V_T \to N \tag{6}$$

N は 0 を含む自然数の集合であり，具体的には

$$r(x)=n \qquad (x \in V_T, n \in N) \tag{7}$$

である*．よって，P は X_0 が節 x から n 個の非終端記号へ分岐する木構造または1個の終端記号に置き換わる文法書き換え規則を与える．

いま，終端記号だけからなる木構造を T_{V_T} で与えると，木構造文法 G_t によって生成される木構造パターンは次式で表せる．

$$L(G_t)=\{\alpha \in T_{V_T} \mid Y \overset{G_t}{\Rightarrow} \alpha, Y \in S\} \tag{8}$$

なお，α は木構造を表している．

一方，木構造文法 G_t に対応する木構造オートマトン M_t は次のような形式で与えられる．

$$M_t=(Q, I, f, F) \tag{9}$$

ここで，Qはオートマトンの状態を示す有限集合

Iはオートマトン M_t の入力記号列

f は Q, I 上で定義された状態遷移関数で次の2つの形式で与えられる．

$\delta(I)=Q$

$\delta(Q, I)=Q$

Fは最終状態を示す集合で，

$F \subseteq Q$．

このような木構造オートマトン M_t によって受理されるすべてのパターンを $T(M_t)$ で表し，$L(G_t)=T(M_t)$ なら，G_t と M_t は等しいという．

さて，ここでパターンを木構造文法によって記述・認識する方法を示す．

まず，木構造文法 G_t を次のように与える．

$G_t=(V, r, P, S)$

$V=\{S, A, B, a, b, \$\}$

$V_t=\{\$, a, b\}$（ただし，\$, a, b はそれぞれプリミティブ・(始点)，→，↑（ベクトル）を与える）

$r(a)=\{2, 1, 0\}$

$r(b)=\{2, 1, 0\}$

$r(\$)=2$

$$P:\ S \to \begin{array}{c} \$ \\ / \ \backslash \\ A \quad B \end{array}$$

$$A \to \begin{array}{c} a \\ / \ \backslash \\ A \quad B \end{array} \qquad B \to \begin{array}{c} b \\ / \ \backslash \\ A \quad B \end{array}$$

$$A \to \begin{array}{c} a \\ | \\ B \end{array} \qquad B \to \begin{array}{c} b \\ | \\ A \end{array}$$

$$A \to a \qquad B \to b$$

G_t によって記述される木構造のひとつは**図-4**のようになる．

いま，図-4で示す G_t に対し，プリミティブ a と b の長さを等しくすると（$|a|=|b|$），G_t は**図-5**に示すようなパターンを記述する．

また，図-4を構文解析するための木構造オートマトンは次のように与えられる．

$M_t=(Q, I, f, F)$

$Q=(q_a, q_b, q_F)$

---

* r は P から一意に決まる．したがって，r を定義しないで，形式 $G_t=(V_N, V_T, P, S)$ で与える方法もある[31]．

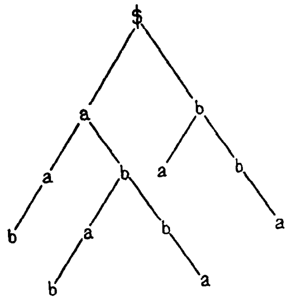


図-4 G_t の木構造

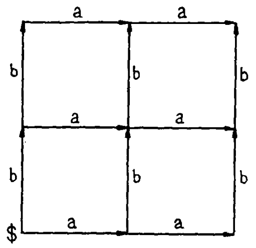


図-5 四角形図

$$I=(a, b, \$)$$

$$\begin{aligned} f:\ &\delta(a)=q_a\\ &\delta(b)=q_b\\ &\delta(q_b, a)=q_a\\ &\delta(q_a, b)=q_b\\ &\delta\{(q_a, q_b), a\}=q_a\\ &\delta\{(q_a, q_b), b\}=q_b\\ &\delta\{(q_a, q_b), \$\}=q_F \end{aligned}$$

3. エラー・パターンに対する修正文法の構成法

文章または言語としてのパターン（一次元記号列または木構造）がある文法によって生成し得るかどうか解釈する手続きを文法的パターン認識法においては構文解析 (Syntax analysis) または文法手続き (Parsing) といった．すなわち，文法$\mathbf{G}$によって正しく構文解析できるパターンxの集合 $L(\mathbf{G})$ は次式で求まった．

$$L(\mathbf{G})=\{x \mid x \in V_T^*, S \overset{G}{\Rightarrow} x\} \tag{10}$$

一般に，nクラスのパターンについて認識しようとすると，まず n 個の文法 $\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_n$ を構成し (Grammatical inference)，それらによってn種類のパターン集合が求まり，

$$L(\mathbf{G}_i)=\{x_i \mid x_i \in V_T^*, S \overset{G_i}{\Rightarrow} x_i\} \quad (i=1, 2, \cdots, n) \tag{11}$$

となる．

ここで，$x=x_1 \cup x_2 \cup \cdots \cup x_n$ である．

このような文法は，理想的なパターンだけを対象としているが，実際には，$\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_n$ のいずれによっても正しく記述できないようなノイズ・パターンあるいは変形パターンが存在する．これらのパターン集合は，一般にエラー・パターンと呼ばれている．いま，エラー・パターンを x' で与えれば，全パターン集合yは

$$y=x \cup x'$$

で表せる．

エラー・パターンは(11)式では記述できないが，理論的には次のような文法を構成すれば記述できることになる．すなわち

$$L(\mathbf{G}_i')=\{x_i' \mid x_i' \in V_T^*, S \overset{G_i'}{\Rightarrow} x_i'\} \quad (i=1, 2, \cdots, n) \tag{12}$$

とすればよいが，(12)式において

$$x'=x_1' \cup x_2' \cup \cdots \cup x_n'$$

で，かつ便宜上 x_i' は $x_i (i=1, 2, \cdots, n)$ と同じクラスに属するエラー・パターンとしている．

結局，すべてのパターンyを記述するためには，次式を導くことになり

$$L(\mathbf{G}_E)=\{y \mid y \in V_T^*, S \overset{G_E}{\Rightarrow} y\} \tag{13}$$

(10)式で与えた文法 $\mathbf{G}$ が $\mathbf{G}_E$ に拡張されることになる．($\mathbf{G} \subsetneqq \mathbf{G}_E, \mathbf{G}_E=\mathbf{G} \cup (\mathbf{G}_1' \cup \mathbf{G}_2' \cup \cdots \cup \mathbf{G}_n')$)

しかし，実際には，エラー・パターンはパターンそのものの“ばらつき”が大きく，各クラス $x_i' (i=1, 2, \cdots, n)$ についてさらに細分化されたパターン $x_{ij}' (j=1, 2, \cdots, n_i)$ が存在しており，これらのパターンをすべて正しく記述できる文法の構成は不可能に近く，仮に構成できたとしても，オートマトンなどによる認識機構が著しく複雑になり，実用性を失う．

したがって，現実的な方法としては，形状などの“ばらつき”をある程度同定することが必要であり，そのために，パターンの不確実な度合(変形度合など)を統計手法の導入によって記述する文法，すなわち**エラー修正文法**の構成が考えられてきた[31)~42)]．このような**エラー修正文法**の多くは，エラー・パターンの不確実さを記述するための基本概念として，次の3つのパターン・プリミティブ・エラー変換 (Syntax error) を設定している[32)~40)]．

① パターン・プリミティブの（不正なパターン・プリミティブとの）入れ替り

② 不要パターン・プリミティブの混入

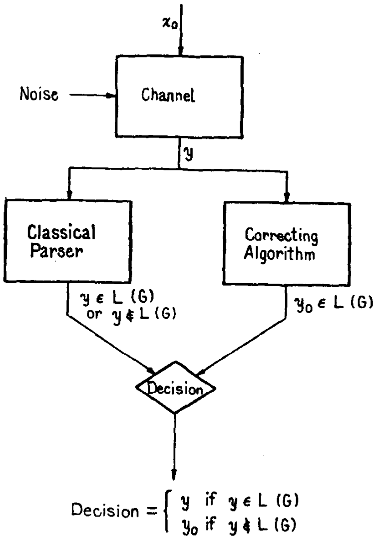


図-6　エラー修正のモデル

③　パターン・プリミティブの欠落

このうち，R. A. Thompson の**エラー修正文法**[39]に対する考え方を**図-6** に示す．この図で左側の流れは従来の方法を示すが，右側は**エラー修正文法**を示し，もとのパターン x_0 がエラー変換され，入力パターン y となり*，これに対する修正パターンが y_0 として求まる**．このような**エラー修正文法**では，パターン y と y_0 との類似度を定量的に定義し，認識理論を構成するが，その場合，類似度を統計的な尺度で定義する方法が一般的である[3]．

すなわち，**エラー修正文法**において，次のような不等式を定義し，すべての $y_0 \in L(G)$ に対し

$$p(y, y_0) \geqq p(y, y_0) \tag{14}$$

または

$$d(y, y_0) \leqq d(y, y_0) \tag{15}$$

の関係を満足するパターン y_0 を選び，それを y の最適修正パターンとする．p, d はいずれも入力パターン y と修正パターン y_0 との類似度を与える関数であり，p は確率，d は距離を定義する．

以下において，**エラー修正文法**を確率関数と距離関数によって構成する代表的な 2 つの方法について紹介する．なお，いずれの方法においても定義している文法 **G** は文脈自由型である．

3.1　確率関数によるエラー修正文法

この方法では，普通図-6 で示すエラー修正に対し，もとのパターン x_0 がノイズなど外的影響によって入力パターン y に変換される確率（図-6 のチャンネル確率でパターン・プリミティブ・エラー変換に起因する）と文法書き換え規則に対して与えられる確率を定義するが，後者の確率のみ定義する方法もある[31]．2 つの確率を用いる場合，任意のパターン x_0 に対し，$P(y|x_0) \cdot P(x_0)$ を計算するが，$P(y|x_0)$ はパターン x_0 が入力パターン y にエラー変換される確率，$P(x_0)$ は $x_0 \in L(G)$ の生起確率で，文法書き換え規則に対して与えられた確率から計算して求まる．このような条件のもとですべてのパターン x_0 に対し

$$p(y|x_0) \cdot p(x_0) \geqq p(y|x_0) \cdot p(x_0) \tag{16}$$

となるような $x_0 \in L(G)$ を入力パターン y に対する修正パターンとして求めるのがここで述べる確率関数によるエラー修正文法の一般的な考え方である*．エラー修正文法に関する研究はいろいろあるが，このうち，M. G. Thomason はチャンネル確率（パターン・プリミティブ・エラー変換確率）を次のように定義し，エラー修正文法を構成している[40]．

①　パターン・プリミティブが入れ替る確率：q_1

②　パターン・プリミティブが混入する確率：q_2

③　パターン・プリミティブが欠落する確率：q_3

したがって，パターン・プリミティブのエラー変換が起らない確率は

$$q_0 = 1 - q_1 - q_2 - q_3 \tag{17}$$

で与えられる．

いま，本稿で定義する文法の書き換え規則がすべて Greibach 標準形に変換されているものとすると[30]，次のように表せる．

$$A \to a\xi. \tag{18}$$

ここで，$A \in V_N$, $a \in V_T$, $\xi \in (V_N \cup V_T)^*$ である．このような条件のもとで，入力パターン y を y_0 に修正するための文法 G_p は次のように構成できる．

$$G_p = (V_N, V_T^{(1)}, V_T^{(2)}, R_P, S). \tag{19}$$

---

* 入力パターン y は理想的なパターン x とエラー・パターン x′ からなる．したがって，図-6 で示すもとのパターン x_0 のうち，一部がノイズなどの影響を受け，エラー・パターン x′ となり，残りが理想的なパターンとなる．

今後，本稿で扱うエラー修正文法への入力パターンは y とする．

** もとのパターン x_0 がパターン・プリミティブ・エラー変換され，入力パターン y となる．したがって，y を逆変換（エラー修正）すれば，修正パターン y_0 が求まる．具体的には，パターン・プリミティブの入れ替りエラーに対し置き換え，混入エラーに対し削除，欠落エラーに対し挿入をすればよい．本稿でのエラー修正文法では，このような考え方が基本となる．

* 以降のエラー修正文法では，y に対する修正パターンとして x_0 でなく y_0 を求めることになる．これはエラー変換をエラー修正に置き換えて考えれば同じことである．

V_N: 非終端記号の有限集合

$V_T^{(1)}$: 入力アルファベット（終端記号）の有限集合

$V_T^{(2)}$: 出力アルファベット（終端記号）の有限集合（修正記号列）

R_P: 確率が定義された文法書き換え規則で，次の形式で与えられる．

$f(r)$: $\Omega \to \gamma, \sigma$

ここで，$\Omega \in V_N$. $\gamma \in (V_N \cup V_T^{(1)})$ は入力記号列，$\sigma \in (V_N \cup V_T^{(2)})$ は出力記号列で，γ が σ に修正出力される確率が $f(r)$ である．

なお，$\Omega \to \gamma, \sigma$ に対する確率 $f(r)$ は文法書き換え規則 $A \to \gamma(=a\xi)$ の生起確率 p と入力記号列 γ が出力記号列にエラー修正される確率 q との積 pq で与えられる．すなわち，$f(r)$ は次のようになる．

① パターン・プリミティブを置き換えて出力する場合，$a_k \in (V_T^{(1)} - \{a\})$ に対し

$$pq_1: \ \Omega \to a\xi, a_k\xi \tag{20}$$

② パターン・プリミティブを削除して出力する場合，$a \in V_T^{(1)}$ に対し

$$pq_2: \ \Omega \to a\xi, \xi \tag{21}$$

③ パターン・プリミティブを挿入して出力する場合，$a_k \in V_T^{(2)}$ に対し

$$pq_3: \ \Omega \to a\xi, \ a_k a\xi \quad \text{または} \quad \to a\xi, \ a a_k\xi \tag{22}$$

これに対し，入力パターンがエラー修正されない確率は(17)式より

$$pq_0: \ \Omega \to a\xi, a\xi \tag{23}$$

となる．

このような確率の適用によって，入力パターンyが y_0 に修正される確率 $p(y, y_0)$ は次式で求まる．

$$p(y, y_0) = f^{(1)}(r) \cdot f^{(2)}(r) \cdots f^{(m)}(r) \tag{24}$$

ここで，$f^{(i)}(r)$ $(i=1, 2, \cdots, m)$ はパターン y から y_0 を導くのに適用した(20)式から(23)式までの確率，m は文法書き換えを行った回数を示す．

したがって，確率関数によるエラー修正文法 G_P から得られる任意の入力パターンyに対する最適修正パターン y_0 は次式で求まる．

$$L(G_P) = \{y_0 \mid p(y, y_0) = \max_{y_0 \in L(G)} p(y, y_0)\} \tag{25}$$

3.2 距離関数によるエラー修正文法

この方法は距離の概念にもとづくクラスタリングによってエラー修正文法を構成する．距離はエラーパターンと正常パターンとの幾何学的な“近さ”を定義できるが，このような距離として，最小二乗距離，ハミング距離，ベイズ距離などが提案されている[34)~38),41),42)]．

このうち，K. S. Fu らの距離関数による**エラー修正文法**においては[34),35)]，まず任意のパターン（パターン・プリミティブ列）$x^{(1)}, x^{(2)} \in V_T^*$ に対し，$x^{(2)} \in T(x^{(1)})$ となるようなエラー修正関数 T を定義する．ここで，T はパターン・プリミティブの置き換え(Replacement)，削除(Deletion)および挿入(Insertion)で文脈自由文法 $G=(V_N, V_T, R, S)$ に対し次のように与えられる．

① 入れ替りエラー修正（置き換え）

$$w_1aw_2 \xrightarrow{T_S} w_1bw_2 \tag{26}$$

② 混入エラー修正（削除）

$$w_1aw_2 \xrightarrow{T_D} w_1w_2 \tag{27}$$

③ 欠落エラー修正（挿入）

$$w_1w_2 \xrightarrow{T_I} w_1aw_2 \tag{28}$$

ただし，$w_1, w_2 \in V_N^*$, $a, b \in V_T^*$ $(a \neq b)$ である．

さて，距離関数によるエラー修正文法の構成法では入力パターンyが y_0 に修正される最小エラー修正回数を2つのパターン y, y_0 に対する距離 $d(y, y_0)$ として定義する．すなわちたとえば，与えられたパターン $y=cbabdbb$ と $y_0=cbbabbdb$ との距離は

$$y = cbabdbb \xmapsto{T_S} cbabbbb \xmapsto{T_S} cbabbdb \xmapsto{T_I} cbbabbdb = y_0$$

となり，最小エラー修正は3回であり，したがって，$d(y, y_0)=3$ となる．

したがって，このような距離の概念を2つのパターンの類似度とすると，入力パターンyと最適修正パターン y_0 との距離は

$$d(y, y_0) = \min_{y_0 \in L(G)} d(y, y_0) \tag{29}$$

となり，(25)式同様，距離関数による**エラー修正文法** G_D から得られる最適修正パターン y_0 は次式から求まる．

$$L(G_D) = \{y_0 \mid d(y, y_0) = \min_{y_0 \in L(G)} d(y, y_0)\} \tag{30}$$

4. 応　用　例

文法的パターン認識法に統計手法を導入した応用例として染色体解析を紹介する．

染色体解析は形状の正確な認識が要求されることから文法的パターン認識法がいち早く適用された[44)]．染

色体は図-3 で示すようなパターンであるが，その切り出し（セグメンテーション）は必ずしも容易でない．また，ノイズの混入もあり，理想的なパターンはむしろ少なく図-3 の 3 つの文法 G_M, G_S, G_A で記述しきれないパターンが多く発生する．

そこで，H. C. Lee らは図-3 の染色体に対し，確率を導入した文法 G_{ST} を構成し（図-7）[31]，未知のパターン y に対し，次の確率計算をし，パターン分類を行った．

$$\mathrm{Max}\ P(\mathrm{y}|i)\cdot P(i) \tag{31}$$

$i=1$ (Median), 2 (Submedian)
3 (Acrocentric)

ここで，$P(\mathrm{y}|i)$ はパターン・クラス i を仮定して，文法書き換え規則の適用により，y を導出した確率，$P(i)$ はパターン・クラスでの生起確率で，2 つの確率積を最大にするパターン・クラス i を求めることによってパターン分類が行われる．

いま，$P(1)=0.26$, $P(2)=0.5$, $P(3)=0.24$ とした場合，**表-1** で示すような確率計算結果を得ているが，このような方法により，パターン・クラスの中間に存在するエラー・パターンを同定できる．

しかし，本質的には，パターン・プリミティブ・レベルでのエラー修正が望ましい．たとえば，図-3 の Median を記述する文法は (19) 式によって，次のよう

$G_{ST}=(V_N, V_T, P, S)$
$V_N=\{S, A, B, D, H, J, E, F, W, G, R, L\}$
$V_T=\{a, b, c, d\}$
（ここで a, b, c, d はそれぞれプリミティブ

を与える）

$P:\ S\xrightarrow{1}AA \quad L\xrightarrow{p_3}FL \quad D\xrightarrow{p_3}FG$
$W\xrightarrow{p_4}d \quad A\xrightarrow{1}cB \quad L\xrightarrow{p_4}HDJ$
$D\xrightarrow{p_3}WE \quad F\xrightarrow{1}b \quad B\xrightarrow{p_1}FBE$
$R\xrightarrow{p_3}RE \quad G\xrightarrow{p_3}FG \quad E\xrightarrow{1}b$
$B\xrightarrow{p_2}HDJ \quad R\xrightarrow{p_4}HDJ \quad G\xrightarrow{1}d$
$H\xrightarrow{1}a \quad B\xrightarrow{p_3}RE \quad D\xrightarrow{p_1}FDE$
$W\xrightarrow{p_2}WE \quad J\xrightarrow{1}a \quad B\xrightarrow{p_3}FL$
$D\xrightarrow{p_2}d$

図-7　確率付き文法 G_{ST}

表-1　確率付き文法の適用結果 ($P(\mathrm{y}/i)P(i)$)

y ＼ i	1(=median)	2(=Submedian)	3(=Acorcentric)
y median	2.6×10^{-6}	1.6×10^{-13}	2.4×10^{-14}
y submedian	6.8×10^{-18}	2.0×10^{-11}	2.0×10^{-13}
y acrocentric	2.0×10^{-27}	4.0×10^{-15}	1.9×10^{-10}

y　y_0
q : →cA, cA
q^2 : →cbB, cbB
$p_3q_0^3$: →cbbB, cbbB
$p_3^3q_0^5$: →cbbbbB, cbbbbB
$p_3p_4q_0^6q_2$: →cbbbbaC, ebbbaC
$p_3^2p_4p_5^4q_0^8q_2$: →cbbbbabbbbC, cbbbabbbbC
$p_3^4p_4p_5^4p_7q_0^4q_2$: →cbbbbabbbbcB, cbbbabbbbdB
$p_3^8p_4p_5^4p_7q_0^{13}q_1q_2$: →cbbbbabbbbbbbbbB, cbbbabbbbdbbbbB
$p_3^8p_4^2p_5^4p_7q_0^{14}q_1q_2$: →cbbbbabbbbbbbbbaC, cbbbabbbbdbbbbaC
$p_3^8p_4^2p_5^7p_7q_0^{17}q_1q_2$: →cbbbbabbbbbbbbbabbbC, cbbbabbbbdbbbbabbbC
$p_3^8p_4^2p_5^7p_7^2q_0^{18}q_1q_2$: →cbbbbabbbbbbbbbabbbcB, cbbbabbbbdbbbabbbbcB
$p_3^{11}p_4^2p_5^7p_7^2q_0^{21}q_1q_2$: →cbbbbabbbbbbbbbabbb cbbbaC, cbbbabbbbdbbbbabbb cbbbaC
$p_3^{15}p_4^4p_5^{11}p_6p_7^2p_8q_0^{35}q_1q_2$: →cabbbbabbbbbbbbbabbb cbbbabbbbdbbbbabbb, cbbbabbbbdbbbbabbb cbbbabbbbdbbbbabbb

図-8　確率関数による染色体パターンのエラー修正

に構成できる．

$G_{PM}=(V_N, V_T^{(1)}, V_T^{(2)}, R_P, S)$
$V_N=(S, A, B, C, D)$
$V_T^{(1)}=V_T^{(2)}=(a, b, c, d)$

R_P は入力記号列 γ を出力記号列 σ に変換する確率付文法書き換え規則集合で，確率 $f(r)$ を構成する文法書き換え規則とエラー修正に対する確率はそれぞれ次のように与えられる．

$p_1=1$: $S\to cA$,　$p_2=1$: $A\to bB$
p_3 : $B\to bB$,　$p_4=B$: aC
p_5 : $C\to bC$,　p_6 : $C\to dB$
p_7 : $C\to cB$,　p_8 : $C\to b$
q_0 : エラー修正をしない確率
q_1 : プリミティブを置き換える確率
q_2 : プリミティブを削除する確率
q_3 : プリミティブを挿入する確率

文法 G_{PM} によって図-3 の Median を記述した例を**図-8** に示す*．この例で，図-3 で示す標準パターン (y_0=cbbbabbbbdbbbbabbbcbbbabbbbdbbbbabbb) に対し，プリミティブ b の混入と b と d の入れ替りのある入力パターン (y=cbbbbabbbbabbbbabbbcbbbabbbbdbbbbabbb) との類似度が (24) 式より確率 $p(\mathrm{y}, \mathrm{y}_0)=p_3^{15}p_4^4p_5^{11}p_6p_7^2p_8q_0^{35}q_1q_2$ として求まる．そして，確率関数 $p(\mathrm{y}, \mathrm{y}_0)$ を (25) 式に適用しを

$$p(\mathrm{y}, \mathrm{y}_0)=\max_{\mathrm{y}_0\in L(G)} p(\mathrm{y}, \mathrm{y}_0)$$

満足する y_0 を y に対する最適修正パターンとして求めることができる．

5. 今後の課題

文法的パターン認識法および統計手法の導入につい

* y から y_0 を導出する過程は部分的に省略されている．

て，その技術動向を展望した．その中で，**文法的パターン認識法**の特長などを述べたが，統計的パターン認識法に比べ，理論を応用していく上でのむずかしさがある．

その主な原因は，次の2点であると考えられる．

① 認識方式が定性的で，かつ認識機構が複雑で装置化がむずかしい．

② 文法の構成としての，いわゆる Grammatical Inference の過程が面倒である．

まず，②の問題については，パターンの特徴をパターン・プリミティブのレベルで抽出し，そのプリミティブ同志の位置関係，順序関係などから，パターンの情報構造を定性的，論理的に解析する関係上，認識理論（回路）が複雑となり，これを現在のコンピュータでそのまま実現しようとすると，ソフトウェアまたはハードウェア機構が著しく複雑となったり，処理時間を要したりして実用性を失う．そこで考えられたひとつの有効な解決法が統計手法の導入であった．しかしこのような方法も応用分野（たとえば，非常に厳密な情報構造の認識を要するような分野）によっては，その有効性が十分に生かされなくなる可能性もあり，必ずしも根本的なパターン認識問題の解決法とはならないかもしれない．したがって，結局は人工知能の研究とも関連して，高度なパターン認識理論が容易に実現できるコンピュータ，すなわち情報を二次元ないし三次元的にとらえ，それらの情報が並列的に超高速で演算でき，かつ人間の思考過程が容易に実現できる連想能力をもつような次世代(または次々世代)コンピュータの出現が待たれるものの，それまでの間は，本稿で述べたような方法によらざるを得ないように思える．

なお，そのほか①に関連して，パターン・プリミティブの自動抽出の問題がある．パターン・プリミティブが正しく，かつ高速に抽出できればエラー・パターンの発生も少なくなり，以降の処理時間が著しく簡単になる．また，全体の処理時間を短縮するために，逐次分類法なども研究されているが[32),24)]，これらはいずれもパターン認識に関する基本的な問題であり，ハードウェア・レベルでの根本的な問題解決が望まれる．

次に，②の問題であるが，**文法的パターン認識法**では，与えられたパターン集合から文法または構造記述法を引き出す，いわゆる文法構成法 (Grammatical Inference) の問題がある．文法構成法は，これが直接パターン認識法の良し悪しに結びつくと言っても過言でないが，その研究はまだ十分に進んでいない．これも先の問題同様，文法構成の面倒な思考過程を代替できるようなハードウェアまたはアルゴリズムなどの開発が待たれる．

以上，**文法的パターン認識法**におけるいくつかの課題について述べたが，**文法的パターン認識法**が統計的パターン認識法までに，その技術応用が普及していくには，いましばらく時間を要すであろう．

しかし，将来先に述べたようなパターン認識に向いたコンピュータでも出現すれば，今より大分事情が異なり，その実用化も進むものと思われる．

謝辞　本稿をまとめるうえで貴重な助言を得た電子総合技術研究所パターン情報部白井良明博士に深謝する．

参考文献

1) Kanal, L.: Patterns in Pattern Recognition: 1968-1974, IEEE Trans. Vol. IT-20, No. 6, pp. 697-722 (1974).
2) Fu, K.S. and Rosenfeld, A.: Pattern Recognition and Image Processing, IEEE Trans. Vol. C-25, No. 12, pp. 1336-1346 (1976).
3) Fu, K.S.: Recent Advances in Syntactic Pattern Recognition, Proc. 4th IJCPR, Kyoto, Japan, (Nov. 7-10, 1978).
4) 日経エレクトロニクス, No. 100: パターン認識の現状と将来, pp. 34-199, 日経マクウヒル社, 東京 (1975).
5) 上坂吉則: パターン認識と学習の理論, ICS ライブラリ 5, 総合図書, 東京 (1971).
6) Special Issue on Syntactic Pattern Recognition, Pattern Recognition Vol. 4, No. 2 (1972).
7) You, K.C. and Fu, K.S.: A Syntactic Approach to Shape Recognition Using Attributed Grammars, IEEE Trans. Vol. SMC-9, No. 6, pp. 334-345 (1979).
8) Ali, F. and Pavlidis, T.: Syntactic Recognition of Handwritten Numerals, IEEE Trans. Vol. SMC-7, No. 7, pp. 537-541 (1977).
9) Bjorklund, C.M.: Syntactic Analysis and Description of Stroke-Based Shapes, PROC. 1977 IEEE Computer Society Conference on PRIP, Troy (June 1977).
10) Jakubowski, R. and Kasprzak, A.: A Syntactic Description and Recognition of Rotary Machine Elements, IEEE Trans. Vol. C-26, No. 10, pp. 1039-1042 (1977).
11) Stockman, G., Kanal, L. and Kyle, M.C.: Structural Pattern Recognition of Carotid Pulse Waves Using A General Waveform Pasing System, Comm. ACM, Vol. 19, No. 12,

pp. 688-698 (1976).

12) Enrich, R. W. and Foith, J. P.: Representation of Random Waveforms by Relational Tree, IEEE Trans., Vol. C-25, No. 7, pp. 725-736 (1976).

13) Mundy, J. L. and Joynson, R. E.: Automatic Visual Inspection Using Syntactic Analysis, PROC. 1977 IEEE Computer Society CPRIP pp. 144-147, Troy (June 6-8, 1977).

14) Fu, K. S.: Grammatical Inference: Introduction and Survey-Part I, IEEE Trans. Vol. SMC-5, No. 1, pp. 95-111 (1975).

15) Fu, K. S.: Grammatical Inference: Introduction and Survey-Part II, IEEE Trans. Vol. SMC-9, No. 4, pp. 409-423 (1975).

16) Earley, J.: An Efficient Context-Free Parsing Algorithm, Comm. ACM, Vol. 13, No 2, pp. 94-102 (1970).

17) Miclet, L.: Inference of Regular Expressions, Proc. Third IJCPR, Calif., U. S. A. (Nov. 8-11, 1976).

18) Horning, J. J.: A Procedure for Grammatical, Information Processing 71, pp. 519-523 (1972).

19) Reghizzi, S. C.: An Effective Model for Grammar Inference, Information Processing 71, pp. 524-529.

20) Fu, K. S. and Bhargava, B. K.: Tree System for Syntactic Pattern Recognition, IEEE Trans., Vol. C-22, No. 12, pp. 1087-1098 (1973).

21) Barrero, A. and Gonzalez, R. C.: A Tree Traversal Algorithm for the Inference of Tree Grammars, Proc. 1977 IEEE Computer Society Conference on Pattern Recognition and Image Processing, Troy, N. Y. (June 6-8, 1977).

22) Brainerd, W. S.: Tree Generating Regular Systems, Information and Control, Vol. 14, pp. 217-231 (1969).

23) Ginsburg, S.: A Mathematical Model of Transformational Grammars, Information and Control, Vol. 15, pp. 297-335 (1969).

24) Chang, N. S. and Fu, K. S.: Parallel Parsing of Tree Languages, Proc. 1978 IEEE Computer Society Conference on Pattern Recognition and Image Processing Chicago (May 31-June 2, 1978).

25) Bhargava, B. K. and Fu, K. S.: Transformations and Inference of Tree Grammars for Syntactic Pattern Recognition, Proc. 1974 IEEE International Conference on Cybernetics and Society, Dallas (Oct. 1974).

26) Brayer, J. M. and Fu, K. S.: A Note the K-Tail Method of Tree Grammar Inference, IEEE Trans. Vol. SMC-7, No. 4, pp. 293-300 (1977).

27) Brayer, J. M. and Fu, K. S.: Application of a Web Grammar Model to an EARTH Resources Satellite Picture, Proc. Third IJCPR, Coronado, Calif. (Nov. 8-11, 1976).

28) Chou, S. M. and Fu, K. S.: Inference for Transition Network Grammars, Proc. Third IJCPR, Coronado, Calif. (Nov. 8-11, 1976).

29) Vigna, P. D. and Ghezzi, C.: Context-Free Graph Grammars, Information and Control Vol. 37, pp. 207-233 (1978).

30) 野崎昭弘, 木村 泉 (Hopcroft, J. E. and Ullman, J. D.): 言語理論とオートマトン, サイエンス社, 東京 (1976).

31) Lee, H. C. and Fu, K. S.: A Stochastic Syntax Analysis Procedure and Its Application to Pattern Classification, IEEE Trans. Vol. C-21, No. 7, pp. 660-666 (1972).

32) Lu, S. Y. and Fu, K. S.: Stochastic Error-Correcting Syntax Analysis for Recognition of Noisy Patterns, IEEE Trans., Vol. C-26, No. 12, pp. 1268-1276 (1977).

33) Thomason, M. G.: Error in Regular Languages, IEEE Trans., Vol. C-23, No. 6, pp. 597-602 (1974).

34) Fu, K. S. and Lu, S. Y.: A Clustering Procedure for Syntactic Patterns, IEEE Trans., Vol. SMC-7, No. 10, pp. 734-742 (1977).

35) Lu, S. Y. and Fu, K. S.: A Sentence,to-Sentence Clustering Procedure for Pattern Analysis, IEEE Trans. Vol. SMC-8, No. 5, pp. 381-389 (1973).

36) Aho, A. V. and Peterson, T. G.: A Minimum Distance Error-Correcting Parser for Context-Free Languages, SIAMJ. Comput., Vol. 1, No. 4, pp. 305-312 (1972).

37) Thomason, M. G. and Gonzalez, R. C.: Syntactic Recognition of Imperfectly Specified Patterns, IEEE Trans. Vol. C-24, No. 1, pp. 93-95 (1975).

38) Wagner, R. A. and Fischer, M. J.: The String-to-String Correction Problem, J. ACM, Vol. 21, No. 1, pp. 168-173 (1974).

39) Thompson, R. A.: Language Correction Using Probabilistic Grammars, IEEE Trans., Vol. C-25, No. 3, pp. 275-286 (1976).

40) Thomason, M. G.: Stochastic Syntax-Directed Translation Schemata for Correction of Errors in Context-Free Languages, IEEE Trans., Vol. C-24, No. 12, pp. 1211-1216 (1975).

41) Filipski, A. J.: A Least Mean-Squared Error Approach to Syntactic Classification, IEEE Trans. Vol. PAMI-2, No. 3, pp. 252-255 (1980).

42) Fung, L. W. and Fu, K. S.: Stochastic Syntax Decording for Pattern Classification, IEEE Trans., Vol. C-24, No. 6, pp. 662-667 (1975).

43) Shaw, A. C.: A Formal Picture Description Scheme as a Basis for Picture Processing Systems, Information and Control, Vol. 14, pp. 9-52 (1969).

44) Ledley, R. S.: High Speed Automatic Analysis of Biomedical Picture, Science, Vol. 146, No. 3641, pp. 216-223 (1964).

(昭和56年1月22日受付)

訂 正

Vol. 22, No. 8, pp. 797～806 掲載の周藤の解説「統計手法を導入した文法的パターン認識法」中の以下の誤りを訂正いたします.

p. 802, 左 ↓ 6

誤：$p(y, y_0) \geqq p(y, y_0)$

正：$p(y, \hat{y}_0) \geqq p(y, y_0)$

p. 802, 左 ↓ 4

誤：$d(y, y_0) \leqq d(y, y_0)$

正：$d(y, \hat{y}_0) \leqq d(y, y_0)$

p. 802, 左 ↓ 3

誤：パターン y_0

正：パターン $\hat{y}_0$

p. 802, 左 ↓ 1

誤：修正パターン y_0

正：修正パターン $\hat{y}_0$

p. 802, 右 ↑ 19

誤：$p(y/x_0) \cdot p(x_0) \geqq p(y/x_0) \cdot p(x_0)$

正：$p(y/\hat{x}_0) \cdot p(\hat{x}_0) \geqq p(y/x_0) \cdot p(x_0)$

p. 802, 右 ↑ 20

誤：$x_0 \in L(\mathbf{G})$

正：$\hat{x}_0 \in L(\mathbf{G})$

p. 803, 左 ↓ 7, 6

誤：最適修正パターン y_0

正：最適修正パターン $\hat{y}_0$

p. 803, 左 ↓ 5

誤：$L(\mathbf{Gp}) = \{y_0 / p(y, y_0) = \max_{y_0 \in L(G)} p(y, y_0)\}$

正：$L(\mathbf{Gp}) = \{\hat{y}_0 / p(y, \hat{y}_0) = \max_{y_0 \in L(G)} p(y, y_0)\}$

p. 803, 右 ↓ 13, 12

誤：最適修正パターン y_0

正：最適修正パターン $\hat{y}_0$

p. 803, 右 ↓ 11

誤：$d(y, y_0) = \min_{y_0 \in L(G)} d(y, y_0)$

正：$d(y, \hat{y}_0) = \min_{y_0 \in L(G)} d(y, y_0)$

p. 803, 右 ↓ 9

誤：最適修正パターン y_0

正：最適修正パターン $\hat{y}_0$

p. 803, 右 ↓ 7, 6

誤：$L(\mathbf{G_D}) = \{y_0 / d(y, y_0) = \min_{y_0 \in L(G)} d(y, y_0)\}$

正：$L(\mathbf{G_D}) = \{\hat{y}_0 / d(y, \hat{y}_0) = \min_{y_0 \in L(G)} d(y, y_0)\}$

p. 804, 右 ↑ 10

誤：$p_4 = B : aC$

正：$p_4 : B \rightarrow aC$

p. 804, 右 ↓ 5

誤：$p(y, y_0) = \max_{y_0 \in L(G)} p(y, y_0)$

正：$p(y, \hat{y}_0) = \max_{y_0 \in L(G)} p(y, y_0)$

p. 804, 右 ↓ 4

誤：y_0

正：$\hat{y}_0$

p. 805, 左 ↑ 10

誤：②の問題

正：(1)の問題

p. 805, 左 ↓ 11

誤：以降の処理時間

正：以降の処理