

帰納学習を用いた図面部品の抽出と分類のための規則の形成

神 嵐 敏 弘[†] 美 濃 導 彦^{††} 池 田 克 夫^{†††}

ベクトル化された論理回路図面からゲートや接続線といった部品を抽出し分類する処理を取り上げ、その処理の過程で用いられる規則を形成する。従来の画像理解分野の規則形成の研究は、対象の形状や色などによる分類、もしくは、図面の記号的な記述だけを対象としたものなどがほとんどであった。これらの研究では、対象の抽出法や記号的な記述が獲得されていることを前提としているが、実際にはこれらの獲得は困難であることが多い。そこで、本研究では、図面から得られた非記号的な記述から、図面部品の抽出と分類の両方を行うための規則を形成することを目標とした。実際の抽出と分類は次の方法で行う。ベクトル化された図面中の任意の線分対が同じ図面部品（同じ種類の図面部品ではない）を構成する確率を求める。この確率を類似度とみなして線分をクラスタリングすることで図面部品を抽出する。次に、抽出された各部品を、それらを構成する線分群の特徴から分類する。上記の確率を計算するために、規則の記述言語（拡張決定リスト）とその形成アルゴリズム（MOKSHA）を考案した。MOKSHA では確率的な結論を出力する規則を形成するために Rissanen の MDL 基準を利用した。これらの手法により、図面部品の抽出と分類がある程度可能であることを実験的に示した。

Rule Formulation Based on Inductive Learning for Extraction and Classification of Diagram Symbols

TOSHIHIRO KAMISHIMA,[†] MICHIIHIKO MINOH^{††} and KATSUO IKEDA^{†††}

This article presents a rule formulation system capable of both extraction and classification of symbols, which indicate gates, wires, etc., in vectorized logic circuit diagram images. This study attempts to create a method to formulate rules for not only classification but extraction of symbols by directly treating sub-symbolic descriptions. Our method for extraction and classification is as follows. First, the probability of a pair of line-segments to be a member of a line-segment set which corresponds to an individual diagram symbol is calculated. By the cluster analyses regarding the probability as similarity measure, the sets of line-segments corresponding to an individual diagram symbol are extracted. These sets of line-segments are classified according to the feature of line-segments in each set. To calculate the probability above, we propose a rule descriptive language (Extended Decision Lists) and a rule formulation algorithm for this language (Algorithm MOKSHA). In order to formulate stochastic rules, our algorithm uses Rissanen's Minimum Description Length Criterion. Our study empirically shows that it is feasible to extract and classify symbols.

1. はじめに

本論文では、論理回路図面からその部品を抽出し、それらの部品を分類するための規則の形成する手法、および、その規則によって部品の抽出と分類を行った

結果の定量的評価について述べる。

ここでいう規則とは、低次の画素などの対象から、より高次の領域などを生成するときに適用するものである。これらの規則は、画像中の対象の種類ごとに異なったものを作成する必要がある、このことが処理対象の種類に依存しない汎用的な画像理解システムの実現を困難にしている。この問題に対する解決法の一つとして、複数のサンプルを与えると、それらから対象に応じた規則を自動的に形成する研究が行われてきた。これらの研究の多くは次の二つに分類できる。

一つは、すでに抽出された対象について形状のモデルを獲得する研究である^{16),17)}。これらの研究では形状

[†] 電子技術総合研究所知能情報部推論研究室

Machine Inference Section, Machine Understanding Division, Electrotechnical Laboratory

^{††} 京都大学工学部高度情報開発実験施設

Integrated Media Environment Experimental Laboratory, Faculty of Engineering, Kyoto University

^{†††} 京都大学工学部情報工学教室

Department of Information Science, Faculty of Engineering, Kyoto University

を分類する対象が抽出されていることを前提としているが、実際にはこの抽出が困難である場合が多い。

もう一つは、Winston の ARCH¹⁹⁾ などのように、記号的な記述を対象とした規則を形成する研究である。これらの研究には、記号的な記述が獲得されていることを前提として、その獲得方法を全く考慮していないか、もしくは、多くのパラメータ調節を必要とする自動的でない方法で記号的な記述を獲得するものが多い。

ここでは、論理回路図面から部品の抽出と分類のどちらか一方ではなく、これら両方を行うための規則を形成する。この規則は非記号的な記述を直接扱う。目標は、サンプルを与えれば、パラメータの調節などを必要とすることなく、これらの規則を自動的に形成することである。

一方、このように複数のサンプルから規則を形成する問題は帰納学習の問題である。帰納学習のためのアルゴリズムには、対象の分類規則を記述するための決定木や決定リストを形成する Quinlan の ID 3^{9),11)} や、対象の間の関係を記述する規則を表す述語論理式を形成する Shapiro の研究¹⁵⁾ などがある。画素などの低次の対象を分類する場合、分類の目的と十分に関連のある特徴をこのような対象だけから得ることは難しいので、他の対象との関係やそれら対象の特徴も考慮した規則が求められる。そこで、対象とそれら関係の両方を扱う規則を記述するために、拡張決定リストと名付ける規則の記述言語とその形成アルゴリズム (MOKSHA*アルゴリズムと名付ける) を提案する。

拡張決定リストは、Quinlan の FOIL¹⁰⁾ にヒントを得て、Rivest の決定リスト¹⁴⁾ や Yamanishi の確率的決定リスト²⁰⁾ に、対象の間の関係についての規則を記述するための拡張を行ったものである。

さらに、図面から得られる非記号的な特徴のみでは、図面部品の抽出や分類を行うには不十分であるので、これらの規則の結論は確率的にしか表現できない。このような確率的な規則を形成するために MOKSHA アルゴリズムでは Rissanen の MDL 基準 (Minimum Description Length Criterion)^{13),21)} を利用した。MDL 基準は、『与えられたデータを、モデル自身の記述も含めて最も短く符号化できるような確率モデルが最良のモデルである』と主張するものである。確率的な規則を必要とするさまざまな問題に、MDL 基準を応用する研究が行われている^{1),2),6),8),12)}。

以下、2章では拡張決定リストと MOKSHA アルゴリズム、3章では図面部品の抽出と分類の具体的な手

法を述べ、4章では実験結果を示して、それらについて考察し、5章では結論を述べる。

2. 拡張決定リストと MOKSHA アルゴリズム

2.1 拡張決定リスト

対象世界の中に N 個の構成要素 E_1, \dots, E_N が存在するとし、各構成要素について $\mathcal{X}_1 \times \dots \times \mathcal{X}_{n_e}$ を定義域とする n_e 次の確率変数ベクトルを定め、構成要素の属性値を表すものとする。また、二つの構成要素 E_i と $E_j (i \neq j)$ の間にある関係が存在する場合 (構成要素の全ての組について関係が存在していなくてもよい)、各関係について $\mathcal{Y}_1 \times \dots \times \mathcal{Y}_{n_r}$ を定義域とする n_r 次の確率変数ベクトルを定め、構成要素間の関係の属性値を表すものとする。構成要素 E_i の k 番目の、 \mathcal{X}_k を定義域とする属性値を $E_i[k]$ 、構成要素 E_i と E_j の間の関係の k 番目の、 \mathcal{Y}_k を定義域とする属性値を $(E_i, E_j)[k]$ と表記する。何番目の属性であるかを表す、この番号 k を属性番号と呼ぶ。図 1 はこの概念図である。

拡張決定リストは、対象世界中の構成要素およびその関係の属性値と、その中から v 個の構成要素の組 E_{i_1}, \dots, E_{i_v} とが指定された場合に、この組に性質 C が存在する確率を出力する規則を記述する。この組は、ある性質 C について定められる命題関数 F の v 個の引数 e_1, \dots, e_v を束縛する。ただし、 F は、引数を束縛する構成要素の組と、対象世界中の構成要素およびその関係の属性値とを入力とし、この組が性質 C を備えているとき 1、備えていないとき 0 をとる関数である。

事例は、 F の引数を構成要素 E_{i_1}, \dots, E_{i_v} で束縛し

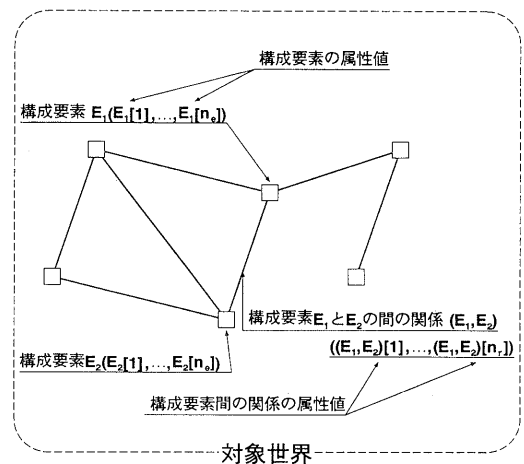


図 1 拡張決定リストの対象とする枠組
Fig. 1 Framework of extended decision lists.

* サンスクリット語で“解脱”の意。

たものと、そのときの関数 F の値の組として与えられ、これを $(F(\dots), \cdot)$ と表記する。

拡張決定リストは、ある整数 m について、順序付けされた条件の列で表されるリスト $DL = \langle t_1, \dots, t_{m-1}, \text{true} \rangle$ とパラメータ $\theta = (\theta_1, \dots, \theta_m)$ で指定される。この意味は、関数 $F(\dots)$ の引数 e_1, \dots, e_v が構成要素の組 E_{i1}, \dots, E_{iv} に束縛されたとき、 t_1, t_2, \dots の順に調べて最初に真になる条件を t_j としたとき、確率 θ_j で $F(E_{i1}, \dots, E_{iv}) = 1$ 、確率 $1 - \theta_j$ で $F(E_{i1}, \dots, E_{iv}) = 0$ が割り付けられることを示している。

条件 t_1, \dots, t_{m-1} は、それぞれ $\exists e_{v+1} \dots \exists e_v [\text{Conjunction}]$ の形で表され、これをタームと呼ぶ。ただし、*Conjunction* とは、構成要素に関する個体変数 $e_i, e_j (i \neq j)$ と属性番号 k で定められる構成要素の属性値 $e_i[k]$ や関係の属性値 $(e_i, e_j)[k]$ についての命題関数 $G(\dots)$ で定められるリテラルの連言であり、 V を $v \leq V \leq V_{\max}$ なる整数、 e_{v+1}, \dots, e_v を個体変数とする。 V の最大値 V_{\max} をいかに定めるかについては 2.3.2 項で述べる。

本論文では、数値属性だけを扱うので関数 G として、 TE_k を \mathcal{X}_k 中の適当なしきい値としたときの $[e_i[k] < TE_k]$ および $[e_i[k] \geq TE_k]$ なる命題関数、 TR_k を \mathcal{Y}_k 中の適当なしきい値としたときの $[(e_i, e_j)[k] < TR_k]$ 、および、 $[(e_i, e_j)[k] \geq TR_k]$ なる命題関数だけを考える。

論理回路図面から実際に獲得した規則は非常に複雑で見にくいので、濃淡画像のある点がエッジの一部であるかを判断する簡単な拡張決定リストの例を示す。画素一つを構成要素 E_i 、その属性 $E_i[0]$ を輝度とする。すべての画素の間に関係が存在するとし、その属性として画素間の距離 $(E_i, E_j)[0]$ や輝度の差 $(E_i, E_j)[1]$ がある場合を考える。このとき、ある構成要素 e_1 がエッジである確率を計算する規則、すなわち、画素 e_1 がエッジの一部であるとき 1、そうでないとき 0 をとる関数 $F(e_1)$ について拡張決定リストの例を図 2 に示す。この図の t_2 は、画素 e_1 から距離 5 未満に画素 e_2 があり、その輝度差が 4 以上のとき、画素 e_1 がエッジの一部である確率は 0.9 であると解釈できる。このように、拡張決定リストは関数 $F(\dots)$ の引数になる構成要素の特徴だけでなく、その他の構成要素の特徴や、それらとの関係を用いて判断する規則を記述できる。

2.2 MOKSHA アルゴリズム

本節では、前節の拡張決定リストに対する学習アルゴリズムである MOKSHA アルゴリズムについて述べる。リテラルの選択のための評価値、パラメータ θ_i の算出式 $Prob(\dots)$ 、および、枝刈りのための手続き

Pruning (DL, θ) については次節で述べ、ここではそれ以外の部分について述べる。

MOKSHA アルゴリズムを図 3 に示す。タームに『カバーされる』とは学習事例の関数 $F(\dots)$ を束縛する物体の組がタームに示された条件を真にすることをいう。図 3 の左に番号を付したアルゴリズムの命令について、次に説明する。

命令 1 は、MOKSHA アルゴリズムに対する入力である。構成要素 E_1, \dots, E_N とこれらの構成要素の間に存在する関係 R_1, \dots, R_N を入力する。学習事例 $(F(\dots), \cdot)$ は、関数 $F(\dots)$ の引数 e_1, \dots, e_v が構成要素

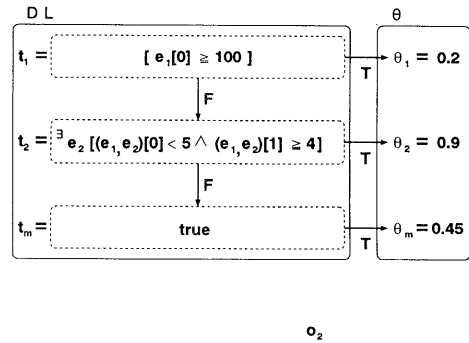


図 2 拡張決定リストの例
Fig. 2 Example of extended decision lists.

```

1 入力
   構成要素  $E_1, \dots, E_N$ 
   構成要素間の関係  $R_1, \dots, R_N$ 
   学習事例  $(F(\dots), \cdot)_1, \dots, (F(\dots), \cdot)_{N'}$ 
   タームの番号  $i \leftarrow 0$ 
   リスト  $DL \leftarrow ()$ , パラメータ  $\theta \leftarrow ()$ 
   現在のリスト  $DL$  にカバーされない事例の集合  $S \leftarrow \{(F(\dots), \cdot)_1, \dots, (F(\dots), \cdot)_{N'}\}$ 
   do
      $i \leftarrow i + 1$ 
     新たに探索するターム  $term \leftarrow \text{true}$ 
     タームの更新回数  $j \leftarrow 0$ 
     探索中のタームにカバーされる事例の集合  $T_i^j \leftarrow S$ 
     do
        $j \leftarrow j + 1$ 
        $T_i^j \leftarrow \{T_i^{j-1} \text{ について評価値を最大にするリテラルを探し、それを } L^j \text{ とする.}$ 
       if (リテラル  $L^j$  の評価値  $\leq 0$ )
          $T_i^j \leftarrow T_i^{j-1}$ 
       else
          $term$  にリテラル  $L^j$  を加える
          $T_i^j \leftarrow \{T_i^{j-1} \text{ に含まれる事例の中で } term \text{ にカバーされる事例}\}$ 
         until (リテラル  $L^j$  の評価値  $\leq 0 \vee T_i^j$  中の事例の関数  $F(\dots)$  の値が全て等しい)
       4 if ( $term$  更新された (=  $term \neq \text{true}$ ))
         パラメータ  $\theta$  に  $\theta_i = Prob(T_i^j)$  をリスト  $DL$  に  $t_i = term$  を追加
          $S \leftarrow S - T_i^j$ 
         until ( $DL$  が更新されなかった  $\vee S$  中の事例の関数  $F(\dots)$  の値が全て等しい)
       5 パラメータ  $\theta$  に  $\theta_i = Prob(S)$  を、リスト  $DL$  に  $t_i = \text{true}$  を追加。
   Pruning( $DL, \theta$ )
   リスト  $DL$  とパラメータ  $\theta$  を出力
  
```

図 3 MOKSHA アルゴリズム
Fig. 3 Algorithm MOKSHA.

に束縛された状態で、かつ、その状態での関数 $F(\dots)$ のとる値が関数と共に組の形で与えられている。

命令2では次の3種類のリテラルから次節で述べる評価値が最も高いリテラル L^j を探索する。ただし、関数 $F(\dots)$ の引数は e_1, \dots, e_v であり、それ以外に *term* には変数 $e_{v+1}, \dots, e_{v'}$ が含まれているとする。

- 1.すでに導入されている変数 $e_1, \dots, e_{v'}$ に関するリテラルで、 $[e[k] < TE_k]$, または、 $[e[k] \geq TE_k]$ の形をした命題関数。
- 2.すでに導入されている変数の間の関係に関するリテラルで、 $[(e_i, e_j)[k] < TR_k]$, または、 $[(e_i, e_j)[k] \geq TR_k] (i \neq j)$ の形をした命題関数。
- 3.すでに導入されている変数と新たに導入する変数 e_{v+1} の間の関係に関するリテラルで、 $[(e_{v+1}, e_i)[k] < TR_k]$, $[(e_{v+1}, e_i)[k] \geq TR_k]$, $[(e_i, e_{v+1})[k] < TR_k]$, または、 $[(e_i, e_{v+1})[k] \geq TR_k]$ の形をした命題関数。

これらの3種類のリテラルについて、変数、閾値、属性番号の、この時点でのすべての可能な組み合わせについて評価する。

命令3では、リテラル L^j の評価値を正にするリテラルが発見された場合に *term* を更新する。*term* は、新たな変数が導入されない場合（前述の1, または、2の場合）と導入される場合（前述の3の場合）とで更新の方法が異なる。

前者の場合、 $term = \exists e_{v+1} \dots \exists e_{v'} [L^1 \wedge \dots \wedge L^{j-1}]$ にリテラル L^j を加えるとき、単にリテラル L^j だけを加えればよいので新たなタームは $term = \exists e_{v+1} \dots \exists e_{v'} [L^1 \wedge \dots \wedge L^{j-1} \wedge L^j]$ となる。一方、後者の場合には新たに導入する変数 e_{v+1} と共に加える必要があり、新たなタームは $term = \exists e_{v+1} \dots \exists e_{v'} \exists e_{v+1} [L^1 \wedge \dots \wedge L^{j-1} \wedge L^j]$ となる。

命令4ではリスト DL とパラメータ θ を更新する。リスト $DL = \langle t_1, \dots, t_{i-1} \rangle$ には $t_i = term$ を追加して $\langle t_1, \dots, t_{i-1}, t_i \rangle$ に、また、パラメータ $\theta = (\theta_1, \dots, \theta_{i-1})$ には $\theta_i = Prob(T_i^j)$ を追加して $(\theta_1, \dots, \theta_{i-1}, \theta_i)$ に更新する。

命令5では生成したリスト DL にまだカバーされていない学習事例をカバーするためのターム **true** とパラメータ $Prob(S)$ とを追加する。

2.3 MDL基準を用いたリテラルの評価値

2.3.1 MDL基準

本研究では、リテラルの評価基準に、有限の大きさの標本からその確率モデルを推定するための基準であるMDL基準 (Minimum Description Length Criterion) を用いている。

MDL基準は、Rissanenによって提案された基準である^{13),21)}。確率的なデータ発生源に対して、ある確率モデルのクラスを予め設定する。このとき、このクラスに情報源の真の確率分布が含まれるかどうかは問題としない。これらの確率モデルのうち、『与えられたデータを、モデル自身の記述も含めて最も短く符号化できるような確率モデルが最良のモデルである』と主張するものである。データおよびモデルの符号化には一意に復号可能な符号を用いる。ただし、ここで言う確率モデルとは何らかの数学的な制約の入ったデータの確率分布のことを指す。

2.3.2 拡張決定リストの符号化

本項では、拡張決定リストの具体的な符号化の方法について述べる。この場合はモデルとはリスト DL とパラメータ θ のことであり、データとは学習事例のことである。ここでは、適用が容易であることから、文献¹⁸⁾の学習事例とパラメータをまとめて符号化する方法を用い、リストだけを別に符号化した。

最初に、あるターム t_i にカバーされる学習事例、および、そのタームのパラメータ θ_i を符号化し、この記述長のリスト DL に含まれるすべてのタームについての総和で、学習事例とパラメータの記述長を表す。

ターム t_i にカバーされる学習事例の集合を T_i とし、そのうち $F(\dots)=1$ となる事例だけからなる集合を T_i^+ とする。また、集合 T の要素数を $|T|$ と記す。文献¹⁸⁾の方法は、パラメータ、すなわち、 $F(\dots)=1$ となる確率の推定量として Laplace の推定量 $\theta_i = (|T_i^+| + 1) / (|T_i| + 2)$ を用いるものであり、学習事例、および、パラメータの記述長の和を

$$Code(a, b) \stackrel{def}{=} \log(a+1) + \log({}_a C_b)$$

で表す。ただし、 $a = |T_i|$, $b = |T_i^+|$ とし、 ${}_a C_b$ はコンビネーションである。また、 \log の底は2とする(以後同様)。このことから前節のパラメータの算出式 $Prob(\dots)$ は上記の Laplace の推定量とする。

次に、リストの符号化を考える。これは、あるタームの記述長について考え、リストに含まれるすべてのタームについて、この記述長の総和をとることで表される。各タームは次のように符号化する。

最初に、このタームが最後のタームであるかどうかを表すために1 bitを必要とする。

次に、このタームに含まれる変数 e_1, \dots, e_v の数 V を符号化する。整数 V を符号化したときの記述長として $\log^* V \stackrel{def}{=} \log(2.865) + \log V + \log \log V + \dots$ (和は正の項に関してのみとられる) が知られているが¹³⁾、ここでは V が小さいので $\log V$ で近似する。

最後に、構成要素とそれらの関係の属性に関するリテラルを符号化するために必要な符号長について考える。\$V\$ 個の変数を含むタームでは、各変数について不等号<または>=をもつリテラルを考えることができ、また、各変数について構成要素の属性の数 \$n_e\$ 個のリテラルが考えられるため、全部で \$U_e=2 \times n_e \times V\$ 個の構成要素に関するリテラルが存在する可能性がある。このうち、実際に \$U'_e\$ 個のリテラルが存在するとき、これらのリテラルは、長さ \$U_e\$ の文字1が \$U'_e\$ 個存在する2進列で表せ、上述の方法により \$Code(U_e, U'_e)\$ だけの記述長で符号化できる。さらに、閾値 \$TE\$ を符号化するために、\$\log \overline{TE} \times U'_e\$ だけの記述長が必要である。ただし、\$\overline{TE}\$ は、閾値の定義域の大きさ \$|X_i|\$ の属性番号 \$i\$ についての平均である。

また、タームで用いられている変数の間の関係は \$(V \times (V-1))/2\$ 個考えることができるので、構成要素の場合と同様に \$U_r=2 \times ((V \times (V-1))/2) \times n_r\$ 個のリテラルを考えることができ、\$Code(U_r, U'_r)\$ の記述長が関係のリテラルを記述するために必要である。閾値の符号化も構成要素に関するリテラルと同様に行うと、ターム一つを符号化するために必要な記述長は \$1 + \log V + Code(U_e, U'_e) + U'_e \times \log \overline{TE} + Code(U_r, U'_r) + U'_r \times \log \overline{TR}\$ となる。

ところで、\$V\$ の最大値 \$V_{max}\$ の設定について、上記の記述長をもとに述べる。MDL基準は、ある確率モデルのクラスの中から最適なモデルを選択する基準であるので、\$V_{max}\$ は予め定める必要がある。タームの記述長は \$V\$ にともない急激に増加するので、新たな個体変数が導入されるために、その増加分以上に学習事例とパラメータの記述長が減少しなくてはならない。学習事例の記述長を十分に減少させるためには、始めに十分な数の学習事例がなくてはならない。よって、学習事例が無限に得られるのでなければ、\$V_{max}\$ を大きな値に設定することに意味はない。また、\$V_{max}\$ を大きくすると規則の獲得に必要な計算量は急激に増加するため、あまり大きな値にはできない。しかし、\$V_{max}\$ を小さくすると、他の構成要素との間の関係も考慮する拡張決定リストの利点が失われる。これらのことを考慮し、本研究では \$V_{max}\$ には \$v+1\$ を用いた。

2.3.3 リテラルの評価値の算出式

MDL基準はモデルの選択基準を与えてはいるが、実際によいモデルを探索する方法についての指針は与えていない。他のMDL基準を用いた帰納学習では次に述べるような方法が用いられている。

最初に、動的計画法などを用いて最良のモデルを求める方法がある⁴⁾。この方法は問題の規模が大きくな

った場合には実用的ではない。

次に、遺伝アルゴリズムや焼きなまし法などのランダム性の高い探索法を用いる方法がある⁹⁾。このような方法は有望だが、まだ探索のためのパラメータの設定の問題が残る。

最後に、ある評価値に基づいて欲張り法による探索でよいモデルを探す方法がある^{7,12)}。これらの研究では、この方法を実際の問題に適用した成果が報告されている。しかし、この方法は評価値の定め方や学習すべき問題の性質によっては、あまりよくない規則しか生成できない場合が考えられるので、評価関数の設定には十分注意する必要がある。

本研究ではこれらのうち最後の手法を採用した。このようなアルゴリズムでMDL基準を用いた決定リスト型の言語に対する研究は見当たらない。そこで、短い記述長をもつ確率モデルを探索するための評価基準として、あるリテラルをあるタームに追加したことによる、そのタームにカバーされる学習事例を符号化したときの、これらの事例の記述長が減少する量を用いる。

$$|T_i^{j+1}| \times \left(\frac{l(T_i^j)}{|T_i^j|} - \frac{l(T_i^{j+1})}{|T_i^{j+1}|} \right)$$

ただし、\$T_i^j\$ は \$j\$ 回目の繰り返しで得られたタームにカバーされる学習事例の集合であり、この集合に含まれる事例を前節の方法で符号化したときの記述長を \$l(T_i^j)\$ とする。

2.3.4 リストの枝刈りの方法

ここでは、タームの枝刈りの手法 Pruning (\$DL, \theta\$) について述べる。

前述の評価関数はタームにカバーされる学習事例だけを評価の対象にしているため、新たなタームを加えても総記述長が減少しない場合がありうる。そこで、タームが真に有用かどうかをもう一度判定し、よりよい確率モデルにするために、不要なタームを削除する操作を枝刈りと呼ぶ。

本学習アルゴリズムでは評価関数によって有用と判定されたタームから順にリスト \$DL\$ に加えられる。よって、記述長の減少に寄与しない不要なタームはリストの最後の方にあると考えられるので、枝刈りはリストの最後の方から行う。

現在の \$DL\$ と \$\theta\$ で指定される拡張決定リストと、ターム \$t_1\$ から \$t_{m-2}\$ までは \$DL\$ と同じだが、長さが一つ短く \$m-1\$ でターム \$t_{m-1}\$ が true であるリスト \$DL'\$、および、パラメータ \$\theta_1\$ から \$\theta_{m-2}\$ までは同じだが、長さが一つ短く \$\theta_{m-1} = Prob(T_{m-1} \cup T_m)\$ であるパラメータ \$\theta'\$ で指定される拡張決定リストとの総記述長を比較

する。

ここで、総記述長は、タームとパラメーターつを記述するために必要な記述長とそのタームにカバーされる学習事例の記述長との、すべてのタームとパラメーターについての総和である。そのため、二つの拡張決定リストの全体について計算をし直す必要はなく、最後の異なるタームについてのみ計算し、比較すればよい。

もしも、後者の拡張決定リストの総記述長の方が短い場合 DL' と θ' を、それぞれ、 DL と θ にする。記述長が小さくならない場合、もしくは、リスト DL 中のタームが一つになった場合に、この手続きを終了する。

3. 図面部品の抽出と分類の手法

3.1 図面部品の抽出と分類

実験で用いた論理回路図面は、AND、OR、および、バッファの論理回路素子 2~4 個と、端子や否定を表す端子、および、接続線で構成される図面で、図 4 に例を示す。このような図面を、8.5 cm 四方の領域に論理回路素子記号の大きさをほぼ揃え、かつ、上下左右のいずれかの方向に論理回路素子が配置されるように手書きした。これらを 50 図面用意し実験に用いた。これらの図面を 150 dpi の精度でイメージ・スキャナを用いて 512×512 画素の 2 値画像として計算機に入力した。

入力された画像に対し、横井の方法³⁾を用いて細線化し、区分的直線近似法により直線近似した。ただし、細線化後の画像と近似された直線の誤差の最大値は 1 画素とした。この画像から、長さが、画像の一辺の長さの 1% 以下、すなわち、5.12 画素よりも短い線分で次の条件を満たす線分を取り除いた。

- ・一方の端点が他の線分と接続しておらず、かつ、もう一方の端点で 2 本以上の線分と接続している線分。
- ・他のどの線分とも接続していない線分。

この処理の結果、残った線分の集まりで表されるデータをベクトル・データと呼ぶ。

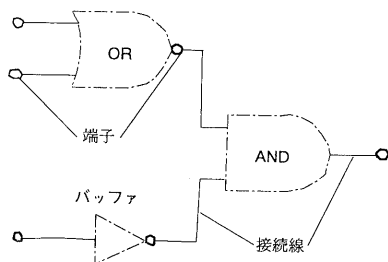


図 4 論理回路図面の例

Fig. 4 Example of logic circuit diagrams.

ベクトル・データ中の図面部品を次の方法で表現する。

- ・同一の図面部品を構成する線分を集めた部分集合を作る。
- ・これらの部分集合を、クラスセットの中の一つのクラスに分類する。

図面部品のクラスセットとして次の 2 通りの場合を用いた。

クラスセット 1 端子、接続線、バッファ、AND、および、OR の 5 種類

クラスセット 2 端子、接続線、および、論理回路素子の 3 種類

図面部品の抽出と分類とは、ベクトル・データから同一部品を構成する線分の部分集合を抽出し、さらに、それらの部分集合を幾つかのクラスに分類することである。これらの操作を、学習用の図面から構成した判断規則を用いて行う。

3.2 図面から抽出した構成要素と関係の属性

拡張決定リストは、対象世界の中に構成要素とこれらの中に存在する関係が定義されている場合を対象とする。構成要素はベクトル・データ中の線分に相当すると考え、学習用に与えられた複数の図面に含まれる全ての線分を同時に扱う。関係は、同じ図面に含まれる全ての線分の組の間に存在すると考え、逆に、別々の図面に含まれる線分の間には関係は存在しないと考える。

構成要素とその関係の属性は 1 次属性と 2 次属性に分けることができる。ここでは、1 次属性について述べ、2 次属性については後で述べる。1 次属性とは論理回路図面のベクトル・データの位置情報や接続情報から直接、獲得できる属性である。どのような 1 次属性を用いるかによって、図面部品の抽出と分類の結果は大きく左右されることが予測される。しかし、本研究では、部品の抽出と分類の手法の開発を主眼に置き、1 次属性には表 1 に示すような、サンプル図面を与えさえすればパラメータ調節を行わず容易に獲得できる

表 1 線分とその関係の属性

Table 1 Attributes of line-segments and of their relations.

線分の 1 次属性	関係の 1 次属性
<ul style="list-style-type: none"> ・線分の 2 つの端点のそれぞれの X 座標と Y 座標 ・線分の midpoint の X 座標と Y 座標 ・その線分が含まれるアークを構成する線分の数 ・線分の長さ 	<ul style="list-style-type: none"> ・ 2 線分が接続しているか ・ 傾きの差 ・ 線分の端点の間の距離 ・ 線分の midpoint の間の距離 ・ 長さの差 ・ 2 線分が同じアークに含まれるか

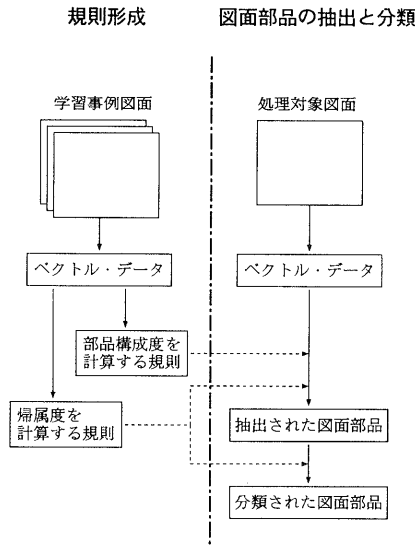


図5 学習事例からの規則形成とそれによる図面部品の抽出と分類
 Fig. 5 Rule formulation from examples, and extraction and classification of diagram symbols by the rule.

出と分類に適用される様子を示す。部品の抽出および分類と、規則形成との大きな相違点は、入力図面を図面理解の対象として扱うか、学習事例として扱うかということ、また、出力結果が、部品が抽出され、さらに、分類された図面となるか、抽出と分類のための規則となるかということであり、ほぼ同様の手続きによって処理される。よって、相違点を明らかにしつつ、まとめて説明する。

全体の手順を図6に示し、各手続きの詳細を次に述べる。3.1節で、図面の構成部品のクラスセットとしてクラスセット1とクラスセット2の2通りを定義した。以下の説明はクラスセット1を用いた場合を中心に述べる。手続き中の数値が、用いるクラスセットによって変化する場合クラスセット2の場合の数値を括弧の中に記す。

1. 線分と関係の1次属性の抽出

ベクトル・データから前節で述べた1次属性を計算する。この段階は抽出と分類、ならびに、規則形成に共通である。

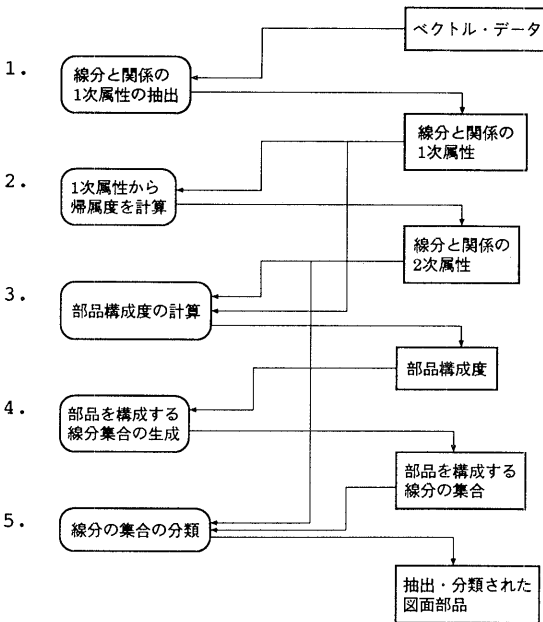
2. 1次属性から帰属性を計算

ベクトル・データの線分は、同じ部品を構成するものどうしでまとめられ、5(3)種類のクラスのうちのいずれかに分類される。そして、各線分が5(3)種類の各クラスに分類される確率を線分の各クラスへの帰属性と呼ぶ。例えば、ベクトル・データ中のある線分が図面中のある“端子”を構成する線分であるかどうかは分からないが、学習事例より得た結果から“端子”に分類される確率は計算でき、この確率をこの線分のクラス“端子”への帰属性と呼ぶ。

学習事例として与えられた図面のベクトル・データから、線分とその間の関係を抽出し、それらの1次属性から5(3)種類の各クラスに対する帰属性を求める拡張決定リストを形成する。ここで、この拡張決定リストの命題関数 $F(e_1)$ は、 e_1 を束縛する線分が指定されたクラスの部品の一部となるときの $F(e_1)=1$ となり、そうでないときに $F(e_1)=0$ となる関数である。

3. 部品構成度の計算

同一図面中の線分の組が、同じ部品(同じ種類の部品ではない)を構成すると予測される確率を部品構成度と呼ぶ。この段階の規則形成とは、線分と関係の属性を用いて部品構成度を求める拡張決定リストを形成することである。この拡張決定リストの関数 $F(e_1, e_2)$ は、 e_1 と e_2 に束縛される線分が同じ部品を構成するとき $F(e_1, e_2)=1$ となり、そうでないときに $F(e_1, e_2)=0$ となる関数である。そして、部品構成度は $F(e_1, e_2)=1$ となる確率で表される。



図中の丸角の箱は抽出と分類の各手続きを、長方形はデータを、そして矢印は各手続きに対する入力と出力を表す。

図6 図面部品の抽出と分類の手順

Fig. 6 Flow of extraction and classification of diagram symbols.

属性を用いた。ただし、アークとは、分岐点や終端点を結ぶひとまとまりの線分のことである。

3.3 図面部品の抽出と分類の手順

図5に、規則形成で形成された規則が図面部品の抽

この部品構成度は線分の帰属度を用いて計算する。この帰属度は1次属性から計算された属性なので、この属性を線分の2次属性と呼ぶ。さらに、線分間の関係について、線分 E_i のあるクラスへの帰属度と E_j のそのクラス帰属度の差の絶対値を2次属性とする。

このように初期的に与えられた属性を処理して新たに合成した属性を参照しても本質的には新たな情報を得たとはいえない。しかし、属性の合成により学習すべき問題が簡単になる。すなわち、より少ない数のチームで構成される拡張決定リストが学習の目的となる場合、合成した属性を用いない場合に比べて、用いた場合の方がより少ない学習事例数で同じ分類誤り率の規則を形成できる。

4. 部品を構成する線分集合の生成

部品構成度を線分間の類似度とみなしてクラスタ分析を行い、同一の部品を構成する線分の集合を抽出する。

クラスタ分析の方法として最短距離法、最長距離法、群平均法などがあるが、ここでは群平均法を用いた。

クラスタ分析は最終的にはすべての分類対象事例を必ず一つのクラスタにまとめてしまう。そこで、図面部品を構成する線分の集合の数、すなわち、クラスタ数が、予め定めた数より小さくなったときクラスタの併合を停止する。ここでは、学習事例中の平均部品数16個になるとクラスタの併合を停止した。

この処理段階は、判断規則を用いないので規則形成の必要はなく、抽出と分類の場合にのみ実行される。

5. 線分の集合を分類

この時点で、同一の部品を構成する線分の集合が取り出されているので、次にこの集合を、いずれかのクラスに分類する。

ある線分の集合に線分 E_1, \dots, E_n が含まれているものとする。これらの各線分には各クラスに対する帰属度が計算されている。線分 E_1 から E_n について、あるクラスに対する帰属度の和をとり、これをすべてのクラスについて行くと、5(3)個の値を得ることができる。この値を最大にするクラスをこの線分の集合が分類されるクラスとする。

この処理段階も、規則形成の必要はなく、抽出と分類の場合にのみ実行される。

4. 実験と考察

ここでは、実際に図面部品を抽出し、分類できることを実験により示す。実験結果を示した表には、平均値と共に、括弧内に標準偏差を記した。

表 2 C 4.5 と MOKSHA の線分の属性だけを用いた線分の分類結果の正解率

Table 2 Mean and standard deviation of accuracies of line-segment classification based on line-segment attributes alone by applying C 4.5 and MOKSHA.

	クラスセット1	クラスセット2
MOKSHA アルゴリズム	.651 (.0309)	.738 (.0288)
C 4.5 アルゴリズム	.610 (.0366)	.738 (.0252)

4.1 MOKSHA アルゴリズムの評価

最初に、MOKSHA アルゴリズムが学習対象の問題の性質を適切に表現した“よい”規則を形成できることを示す。

構成要素を、それらの属性によって分類する規則を形成する代表的な ID 3 型のアルゴリズム一つある C 4.5¹¹⁾ と MOKSHA アルゴリズムとを比較する。比較のために、拡張決定リストの、構成要素間の関係を扱うための拡張部分を削除して、すなわち、 $V_{\max} = v$ として、実験した。

3.3 節の手順 2 の帰属度を計算する規則を、クラスセットに含まれる各クラスについて形成する。この規則を用いて計算された帰属度を最大にするクラスに、線分は分類されるものとする。そして、分類されるべきクラスに正しく分類される線分の数の割合によって、これら二つのアルゴリズムを評価した。

3.1 節のクラスセット 1 と 2 それぞれについて、50 枚の図面を、5 枚ずつ 10 個のグループに分け、9 個のグループ、45 枚の図面を学習事例として規則を形成し、残りの 1 グループ、5 枚の図面に含まれる線分が正しく分類される割合を求める試行を 10 回行い、それらの平均を表 2 に示す。また、C 4.5 アルゴリズムのパラメータはデフォルトの値を用いた。

クラスセット 1 の場合、C 4.5 の正解率は MOKSHA の場合の正解率から標準偏差を引いた値よりも小さく、MOKSHA の方が優れている。クラスセット 2 の場合は、平均値は同じであり、二つのアルゴリズムの間には有意な差があるとはいえない。これらの結果を総合すると、MOKSHA アルゴリズムがよい規則を形成できるアルゴリズムであるといえる。

4.2 線分間の関係を考慮した規則の効果

拡張決定リストの、ある構成要素だけでなく、他の構成要素やそれらとの関係も考慮した規則の効果について実験した。

線分の属性だけではなく、3.2 節で述べた関係が図面に存在するものとし、これらの属性も利用して帰属度を計算した。実験は、4.1 節と同じ方法で行った。結

表 3 関係の属性も用いた線分の分類結果の正解率

Table 3 Mean and standard deviation of accuracies of line-segment classification based on both attributes and relational ones of line-segment.

クラスセット 1	クラスセット 2
.655 (.0283)	.769 (.0214)

果を表 3 に示す。ただし、 V_{\max} を $v+1$ である 2 とした。

表 2 と比較すると、平均値は本実験の方が優れており、線分の関係についても考慮することの有効性が示されている。特に、クラスセット 2 の場合、二つの平均値には、標準偏差よりも大きい有意な差がある。帰属度の計算の対象である線分の属性だけでなく、他の線分との関係やそれらの属性も考慮することによって、帰属度が改善されている。他の線分について考慮することの効果は、2.3.2 項でも述べたように、学習事例の数に大きく依存しているため、さらに多くの学習事例があればこの効果が大きくなることが期待される。

4.3 部品構成度の計算

ここでは、3.3 節の手順 3 の部品構成度を求め、その性質について考察する。

部品構成度は、関係が存在する線分対の間に定義される。実験は、3.2 節のように、同じ図面を構成するすべての線分対の間に関係が存在する場合のほか、次に述べる関係だけが存在する場合についても実験した。

同じ図面を構成する線分対で、それらの端点間の距離が 16 画素以下である場合にのみ、関係が存在する。

前者を全体実験、後者を近傍実験と呼ぶ。

さらに、これら二つの実験それぞれの場合について、次の三つの場合の実験を行った。

1 次属性実験

1 次属性だけをを用いた場合

クラスセット 1 実験

クラスセット 1 についての帰属度を 2 次属性とし、1 次と 2 次の両方の属性を用いた場合

クラスセット 2 実験

クラスセット 2 についての帰属度を 2 次属性とし、1 次と 2 次の両方の属性を用いた場合

ただし、2 次属性は 4.1 節で形成した規則を用いて求めた。

4.1 節と同様、9 グループ、45 枚の図面で規則を形成し、1 グループ、5 枚の図面の部品構成度を計算す

表 4 部品構成度の平均値

Table 4 Mean of probabilities that a pair of line-segments consists of a diagram symbol.

		同じ部品	違う部品	平均値の差
全体 実験	1 次属性実験	.577 (.1897)	.377 (.2030)	.200
	クラスセット 1 実験	.584 (.2024)	.401 (.1686)	.183
	クラスセット 2 実験	.645 (.2268)	.462 (.2198)	.183
近傍 実験	1 次属性実験	.708 (.2477)	.435 (.1986)	.273
	クラスセット 1 実験	.689 (.2445)	.362 (.2557)	.327
	クラスセット 2 実験	.757 (.2337)	.408 (.3161)	.349

る。これを 10 回行って、すべての図面について部品構成度を計算する。 V_{\max} はできるだけ大きく設定する方が望ましい。しかし、部品構成度の計算では引数となる構成要素が二つであり、さらに、線分対の数は線分の数の数百倍であるため、 $V_{\max}=v$ としても前節の実験以上の計算量が必要となる。よって、 $V_{\max}=v$ として部品構成度を計算せざるを得なかった。

図面中の関係が存在するすべての線分対を、真に同じ図面部品を構成するものと、構成しないものに分け、それぞれの線分対の集合について部品構成度の平均を表 4 に示す。

部品構成度は、その定義から、真に同じ図面部品を構成する線分対では 1.0 に、構成しない線分対では 0.0 に近い方がよりよい規則が形成されているといえる。6 通りのいずれの場合も、同じ図面部品を構成する場合の方が、そうでない場合よりも部品構成度の平均値は大きい。しかし、二つの平均値の差と標準偏差をみると、これら二つの場合を部品構成度によって完全には分離できないことがわかる。このことは、次にこの部品構成度を用いて図面部品の抽出を行う際に問題となった。

全体実験と近傍実験を比較して、次のようなことがいえる。全体実験の場合の方が、近傍実験の場合よりも、同じ図面部品を構成する場合とそうでない場合の部品構成度の差が小さい。このことにより、全体実験の場合は、多くの線分対について有意な規則性を見つけてことができず、望ましくない規則が形成されているといえる。『画像理解では、空間的な近傍が重要である』という知識を用いることによって、近傍実験では、全体実験よりも問題が単純になっているため、よい規則を形成することができている。

1 次属性実験、クラスセット 1 実験、および、クラスセット 2 実験の三つの場合を比較した。表 4 の結果から、全体実験では、1 次属性実験の方が、近傍実験では 1 次属性実験、クラスセット 1 実験、クラスセッ

ト2実験の順に平均値の差が大きくなっており、あまりはっきりした傾向は見られなかった。

本研究の手法で得られた部品構成度は線分対が同じ部品を構成するかどうかを判別するために利用可能であるが、完全に判別できるものとはならなかった。これは、次の二つの問題点によると考える。第一に、部品構成度を計算するとき、ベクトル・データから容易に獲得できる1次属性を適当に選択したが、この選択は部品構成度に大きく影響する。線分の長さや位置などの基本的な特徴だけでなく、より問題に適した様々な特徴も利用すれば部品構成度は改善できると考える。第二に、計算の対象の構成要素だけでなく、その他の構成要素も考慮した規則を記述できるように拡張決定リストを考察したが、部品構成度の計算においては計算量の問題から $V_{\max} = v$ とせざるを得ず、二つの線分について考慮した規則しか形成できなかった。より少ない計算量で、より多くの構成要素について考慮できるように、規則の記述言語やその形成アルゴリズムを改良する必要がある。

4.4 図面部品の抽出

ここでは、3.3節の手順4の方法で、前節の6通りの部品構成度を利用したクラスタリングによって図面部品を抽出した。クラスタリングの結果の評価方法には確立された方法がないので、以下のように3通りの方法で定量的評価を試みた。

前節の6通りの部品構成度について、クラスタリングの結果、線分対が、真に同じ部品を構成しているかどうかを正しく判定される正解率を各図面ごとに求め、この割合の図面50枚についての平均を示す。クラスタリングはクラスタの数が、予め定めた数より小さくなったときに停止する。この数を、3.3節で述べた、学習事例中の平均部品数の16個とする代わりに、5、10、20、30個とした場合について実験した結果を表5に示す。

一つの線分が一つのクラスタを構成するとした場

合、すなわち、いかなる線分対も同じ部品を構成しないとした場合の正解率0.916より劣っており、全く学習の効果ない結果が幾つか見られるが、これは次のような理由によるものと考えられる。

三つの線分が同じ部品を構成する場合には、これらの間の三つの線分対のうち二つが同じ部品を構成し、残りひとつが同じ部品を構成しないということはありえない。このような制約を利用することで、部品構成度が十分に正しく計算できている場合には正解率はクラスタリングによって改善できる。しかし、逆に部品構成度があまり正しく計算できていない場合は正しく判定された線分対の結果を書き換える場合が生じ、正解率は下がる。このため、全く学習の効果のない結果が生じたと考える。

部品構成度は近傍実験の方が全体実験よりも優れていることを前節で述べたが、このため、全体実験では学習の効果が認められていないが、近傍実験ではその効果がみられる。

前述の正解率では、正解率の下限.916と比較して、学習の効果があつたかどうか程度は評価できるが、各実験の間の結果の差があまり見られずそれ以上の評価はできない。そこで、二つの確率分布が一致している度合の尺度として代表的なKullback-Leiblerのダイバージェンスを用いて評価した。

真に同じ部品を構成する線分対の割合を P 、本手法により同じ部品を構成すると判断された線分対の割合を Q で表すと、ダイバージェンスは $P \times \log(P/Q) + \bar{P} \times \log(\bar{P}/\bar{Q})$ で表される。この量は、予測された確率分布が真の分布と一致する場合にのみ0になり、分布の相違が大きくなるにつれ大きくなる。この値の50図面に対する平均を表6に示す。

この実験は、表5の実験の正解率が高くてもダイバージェンスの低い場合があり、正解率が学習の効果を完全に表しているとは一概にはいえないことを示している。学習の効果の評価にはまだ改善の余地があると考えている。また、近傍実験の方が、全体実験より

表5 線分対の分類結果の正解率

Table 5 Mean of accuracies of line-segment pair classification.

最大クラスタ数		05	10	16	20	30
全体 実験	1次属性実験	.696	.778	.785	.790	.864
	クラスセット1実験	.643	.771	.796	.819	.889
	クラスセット2実験	.607	.683	.749	.795	.873
近傍 実験	1次属性実験	.819	.909	.929	.933	.932
	クラスセット1実験	.775	.914	.931	.933	.933
	クラスセット2実験	.739	.912	.930	.933	.933

表6 Kullback-Leiblerのダイバージェンス

Table 6 Kullback-Leibler divergence.

最大クラスタ数		05	10	16	20	30
全体 実験	1次属性実験	.304	.179	.174	.174	.082
	クラスセット1実験	.496	.230	.164	.122	.072
	クラスセット2実験	.546	.346	.229	.137	.056
近傍 実験	1次属性実験	.117	.007	.010	.021	.050
	クラスセット1実験	.214	.010	.009	.021	.050
	クラスセット2実験	.289	.011	.010	.022	.052

表 7 相互情報量
Table 7 Mutual information.

最大クラスタ数		05	10	16	20	30
全体 実験	1次属性実験	.138	.168	.164	.171	.197
	クラスセット1実験	.124	.165	.175	.173	.187
	クラスセット2実験	.095	.106	.126	.128	.137
近傍 実験	1次属性実験	.291	.263	.231	.223	.189
	クラスセット1実験	.322	.320	.247	.227	.197
	クラスセット2実験	.297	.305	.242	.225	.189

も優れた結果が得られることは、この実験でも確かめられる。さらに、最大クラスタ数が16の場合にダイバージェンスは小さくなっており、このことから、クラスタリングを停止するクラスタ数は、学習事例中の平均部品数としたことは適当である。

最後に、実際に学習によってどれくらいの情報量が獲得できたのかの目安になる実験を行った。真に線分対が同じ部品を構成しているかどうかという事象を A 、本手法により判定した結果の事象を B で表す。また、 $I(A; B)$ を相互情報量、 $H(A)$ をエントロピーとして、 $I(A; B)/H(A)$ なる量を計算した。この量は、線分対 E_1E_2 と E_1E_3 は同じ部品を構成するが、線分対 E_2E_3 は同じ部品を構成しないということはないという拘束条件を考慮していないが、図面部品の抽出に必要な情報量 $H(A)$ のうち、どれくらいの割合の情報を本手法によって獲得できたかを知る指針にはなる。この値の50図面に対する平均を表7に示す。

この実験の結果、最も良好な結果の場合では30%ほどの情報が獲得されているといえる。また、近傍実験の方が良好な結果が得られることは、この実験の結果からも示されている。

実際の、図面部品の抽出の結果の一つを図7に示す。同じ部品に属する線分は同じ線種で表した。これは、近傍実験、クラスセット2実験、最大クラスタ数16の場合の正解率が.948である例である。

図7では、抽出されるべき集合と、クラスタリングの結果の差異は大きい。これは、次の二つの理由によると考える。第一に、前節で求めた部品構成度では、線分対を同じ部品を構成するものとそうでないものに十分に分離できなかったからである。第二に、ここでは、群平均法など既存のクラスタリングの手法を用いた。しかし、部品構成度には線分の組 E_1E_2 や E_1E_3 は非常に類似しているが、 E_2E_3 は全く類似していないといった、類似度の間に矛盾がある場合が多い。このような状況を考慮したクラスタリング手法を考案する必

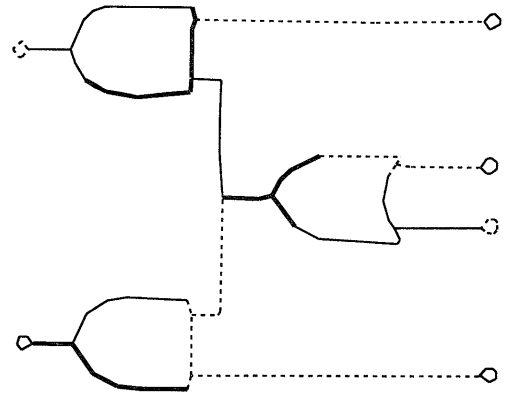


図7 本手法による部品抽出結果の例
Fig.7 A result of extraction of diagram symbols by our method.

表 8 図面部品の分類の正解率
Table 8 Mean and standard deviation of accuracies of symbol classification.

クラスセット1	クラスセット2
.834 (.1216)	.891 (.0876)

要があると考えられる。

4.5 図面部品の分類の実験

ここでは、3.3節の手順5で述べた手法で図面部品の分類が可能であることを示す。

前節の方法では、図面部品は、本節の分類の前処理として利用できるほど正確に、図面部品を抽出できなかった。そこで、本節では前節の結果を用いずに、正しく抽出された図面部品の集合を分類する実験を行う。

実験は、3.3節の手順5の手法で各図面部品を分類し、4.1節と同様に、正しく分類された線分の割合を求めた。ただし、この割合は図面1枚ごとに求め、これを50枚の図面について行った。結果を表8に示す。

クラスセット1と2のいずれの場合も、4.1節の場合(表2参照)よりも平均値で優れており、さらに、その差は標準偏差よりも大きい。これらのことから、部品の抽出が正しく行えるならば本研究の手法により図面部品を分類することができるといえる。

5. おわりに

本論文では、論理回路図面からその部品を抽出し、それらを分類するための規則の形成と、その規則による図面部品の抽出および分類について述べた。

図面部品の抽出と分類どちらか一方だけでなく、両方を行うための規則を形成し、この規則は非記号的な

対象を直接扱い、その形成の過程を自動化することを目標とした。このため、拡張決定リストと MOKSHA アルゴリズムを考案し、これらを用いた図面部品の抽出と分類を行う手法を提案した。

本研究の手法によって、記号的な記述に対する規則ではなく、低次の構成要素の非記号的記述に対する規則を形成できることを示した。この形成過程で V_{\max} と近傍を定める距離以外のパラメータを用いることなく、サンプルを与えさえすれば低次の構成要素から自動的に、部品を抽出し分類することができた。しかし、抽出すべき部品と、低次の構成要素の非記号的特徴との間の隔たりは大きく、部品と非記号的特徴の間の対応を十分に見出すことができなかつた。今後は、線分だけでなくアークやループといった、自動的に抽出できるもう少し高次の特徴を利用したり、規則の記述言語やクラスタリング手法を改良したりすることでこの隔たりを縮めたい。

参 考 文 献

- 1) Derthick, M.: A Minimal Encoding Approach to Feature Discovery, *AAAI-91*, (1991).
- 2) 顧 海松, 浅田 稔, 白井良明: 動き情報に基づくエッジセグメントの最適分割, 信学論 D-II, Vol. J76-D-II, pp. 1544-1553 (1993).
- 3) 長谷川純一, 輿水大和, 中山 晶, 横井茂樹: 画像処理の基本技法<技法入門編>, 技術評論社, (1986).
- 4) 伊藤秀一: MDL のパターン認識への応用, 人知誌, Vol. 7, pp. 608-614 (1992).
- 5) 小長谷明彦: 確率的アプローチによる遺伝子情報処理, 人知誌, Vol. 8, pp. 427-438 (1993).
- 6) Leclerc, Y. G.: Constructing Simple Stable Descriptions for Image Partitioning, *International J. of Computer Vision*, Vol. 3, pp. 73-102 (1989).
- 7) 中莖洋一郎, 古関義幸, 田中みどり: 確率モデルの学習方式と診断への応用, 人知誌, Vol. 7, pp. 862-869 (1992).
- 8) Pednault, E. P. D.: Some Experiments in Applying Inductive Inference Principles to Surface Reconstruction, *11th IJCAI* (1989).
- 9) Quinlan, J. R.: Induction of Decision Trees, *Machine Learning*, Vol. 1, pp. 81-106 (1986).
- 10) Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, Vol. 5, pp. 239-266 (1990).
- 11) Quinlan, J. R.: *C4.5 Programs for Machine Learning*, Morgan Kaufmann (1993).
- 12) Quinlan, J. R. and Rivest, R. L.: Inferring Decision Trees Using the Minimum Description Length Principle, *Information and Computation*, Vol. 80, pp. 227-248 (1989).
- 13) Rissanen, J.: A Universal Prior for Integers and Estimation by Minimum Description Length, *The Annals of Statistics*, Vol. 11, pp. 416-431 (1983).
- 14) Rivest, R. L.: Learning Decision Lists, *Machine Learning*, Vol. 2, pp. 229-246 (1987).
- 15) Shapiro, E. Y.: An Algorithm that Infers Theories from Facts, *7th IJCAI* (1981).
- 16) 高木宣明, 美濃導彦, 坂井利之: 正・負シンボルを用いた論理型図面の図面シンボル認識, 信学技報, Vol. PRL 85-82, pp. 49-56 (1986).
- 17) 上田修功, 鈴木 智: 凹凸構造の一般化に基づく輪郭形状モデルの自動獲得, 信学論 D-II, Vol. J74-D-II, pp. 220-229 (1991).
- 18) Wallace, C. S. and Patrick, J. D.: Coding Decision Trees, *Machine Learning*, Vol. 11, pp. 7-22 (1993).
- 19) Winston, P. H. (著), 白井良明, 杉原厚吉 (訳): コンピュータービジョンの心理, 5. 例を示すことにより構造的な記述を学習する, pp. 173-235, 産業図書 (1979).
- 20) Yamanishi, K.: A Learning Criterion for Stochastic Rules, *Machine Learning*, Vol. 9, pp. 165-203 (1992).
- 21) 山西建司, 韓 太舜: MDL 入門: 情報理論の立場から, 人知誌, Vol. 7, pp. 427-434 (1992).
(平成 6 年 7 月 4 日受付)
(平成 7 年 1 月 12 日採録)

神 嶋 敏 弘 (正会員)



昭和 43 年生。平成 4 年京都大学工学部情報工学科卒業。平成 6 年同大学院修士課程修了。同年電子技術総合研究所入所。機械学習とその応用の研究に従事。人工知能学会会員。

美 濃 導 彦 (正会員)



昭和 53 年京都大学工学部情報工学科卒業。昭 58 年同大学院博士課程修了。同年工学部助手。昭和 62 年～63 年マサチューセッツ州立大学客員研究員。平成元年京都大学工学部附属高度情報開発実験施設助教授。画像処理、人工知能、知的コミュニケーション関係の研究に従事。工学博士。IEEE, ACM, 画像電子学会, 電子情報通信学会, ロボット学会各会員。

**池田 克夫 (正会員)**

昭和 35 年京都大学工学部電子卒業. 昭和 40 年同大学院博士課程単位取得退学. 同年, 京都大学助手. 昭和 46 年京都大学助教授. 昭和 52 年 5 月筑波大学教授. 昭和 63 年 8 月京都大学教授. この間, 昭和 46 年 9 月~昭和 47 年 3 月米国ユタ大客員研究員. 同年 3~8 月マサチューセッツ工科大客員研究員, 昭和 59 年 10~11 月スイス連邦工科大客員研究員. 高度の情報処理システムの構成に興味を持ち, コンピュータネットワーク, 分散処理システム, マン・マシンインタフェース, 画像理解の分野で研究を行っている. 著書に, コンピュータユーティリティの構造(昭晃堂), オペレーティングシステム論(本学会), データ通信(昭晃堂)などがある. 工学博士. 人工知能学会, IEEE, ACM 各会員, IPL editor.
