

4 Q-8 ヴィジュアルパターンからの記号的制約表現の学習*

開 一夫[†] 安西 祐一郎[‡]
慶應義塾大学[§]

1 はじめに

我々は、これまで、ヴィジュアルパターンから記号的表現を学習することを目的としたシステム ACORN を設計・実装してきた [1][2]。ACORN は、ヴィジュアルパターンから抽出した対象間の関係属性を用いて、対象間の空間的關係を制約論理型言語のプログラムとして学習する。本稿では、まず、ACORN を概観した後、ACORN で用いている学習アルゴリズムについて述べる。このアルゴリズムは、インクリメンタルに入力される概念の正事例・負事例に対して、選言を含む制約プログラムを出力するものである。

2 学習システム Acorn

ACORN は、Perceptual Module, Constraint Generator そして Performance Module の3つのモジュールより構成される。Perceptual Module はラスタイメージから数値的なシーンの記述を生成する。Constraint Generator は Perceptual Module から生成された数値的記述と教師より与えられる対象間の空間的關係をあらわすシーンの記号的な記述から、空間的關係の定義を制約論理型言語 EPOCH[3] のプログラムとして学習する。Performance Module は、学習された制約プログラムを用いて、空間的關係間の推論やレイアウトの生成、シーンの記号的記述といったパフォーマンス・タスクを実行する。

例えば、図1(a)に示したような“顔”のイメージと、“顔”を構成する各部分間の空間的關係を表す述語表現(図1(b))が与えられると、Constraint Generator は、各々の空間的關係に対する EPOCH のプログラムを生成する。この例の場合、学習された EPOCH のプログラムを用いることで、ある部分が欠如した不完全な“顔”から全体の修復を行ったり(図2(a))、空間的關係の記号的記述から各部分を配置する(図2(b))といったパフォーマンス・タスクを行なうことができる。

学習した概念の表現言語として EPOCH を用いた最大の利点は、学習結果を様々なパフォーマンス・タスクに利用できる点にある。一般に、制約論理型言語では、変数の値を求めるための手続きを陽に記述する必要がなく、変数間の関係を宣言的に記述するのみでよい。EPOCH では、こうした特徴に加え、各々の制約式に対して、必須制約と目標制約という2種類の解釈のいずれかを与えることができる。必須制約は、CLP(ℝ)に代表される既存の処理系における制約の解釈と同じく、“必ず充足されなければならない制約”である。一方、目標制約は、既存の処理系では扱うことのできなかった“充足される必要のない”弱い意味づ

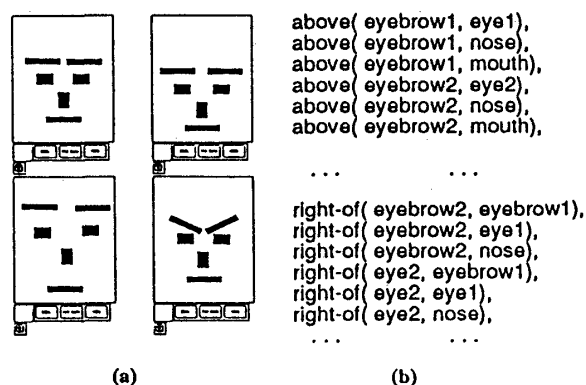


図1: ACORN における“顔”の事例: (a) 画像イメージ (b) 空間的關係の述語表現(一部)

けの制約である。目標制約の導入により、必須制約を充足する解が複数存在しても、目標制約の達成の度合により最適解を決定できるようになる。EPOCH は、プログラム中に記述された制約式の集合を標準形の線形計画問題に変換する機能を備え、制約解消系がシンプレックス法を用いることでその最適解を求めている。

一般にレイアウト問題のような制約充足型の問題では、制約条件が不等式を用いて表現される場合が多い。ACORN では、これらの不等式制約の集合から具体的事例の生成を行なう際に、制約の典型値を EPOCH の目標制約として表現することにより、空間的制約を満足する空間的对象の存在可能な範囲から、具体的な位置を決定している。

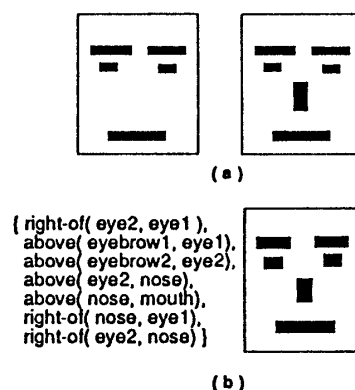


図2: パフォーマンス・タスクの例: (a) “鼻”のない顔と修復された“顔”. (b) 空間的關係の記号的記述とそこからレイアウトされた“顔”.

*Learning Constraint Expressions from Visual Patterns

[†]Kazuo HIRAKI

[‡]Yuichiro ANZAI

[§]Keio University

3 学習アルゴリズム

ACORN における学習で、その中心となるアルゴリズムは区間への一般化と呼ばれ、空間的関係の正事例・負事例から区間を生成する。このアルゴリズムは、負事例の属性値を積極的に扱い、インクリメンタルに入力される事例から選言的な区間制約を出力する。ここで、区間制約は EPOCH の線形不等式を用いて表される。表 1 は、このアルゴリズムの概要を示したものである。表 1 では、区間制約の集合 R_n が存在する場合に、事例 I が入力されたときの振舞いが示されている。

表 1: ACORN の学習アルゴリズム区間への一般化の概要

```

For a set of Region  $R_n$  and instance  $I$ 
  If  $I$  is positive instance
    If exist  $r \in R$  that contains  $I$ 
      then do nothing
    Else change a Region  $r \in R$ ,
      extend  $r$  so that it includes  $I$ .
  If  $I$  is negative instance
    If exist  $r \in R$  that contains  $I$ 
      then replace  $r$  with boundary points
    Else do nothing
  
```

図 3(1)–(5) は、事例の入力に伴い、本アルゴリズムで生成される制約の変化を、空間的関係 *next-to* の場合を例に取って示したものである。ここでは、簡単のため空間的対象を点として扱っている。まず、すでに (1) の斜線で示される領域に対応した区間制約が生成されているものとする。次に (2) の b_1 で示される正事例が入力されると、すでにあった制約は (3) の斜線で示される領域に拡張される。次に b_2 で示される負事例が入力されると、それを含む領域は (4) のように領域の端点で置き換えられる。

以上のようにして、事例が逐次入力されるごとに、その時点で生成されている領域に対応した EPOCH の制約プログラムが生成される。例えば、図 3(5) に対して、以下のようなプログラムが生成される。ここで、“*pref*:(Expr)” は Expr が目標制約であることあらわす。図 3(5) では、領域が 2 つ生成されているので、これに対応する EPOCH のプログラムも、選言を用いてあらわされている。

```

constraint(next_to,
  loc(A,[X1,Y1]), loc(B,[X2,Y2])):-
  X = X2 - X1, Y = Y2 - Y1,
  Y <= -0.345895*X,
  0.361615*X <= Y,
  pref:(X = 7.5), pref:(Y = 0.0).
constraint(next_to,
  loc(A,[X1,Y1]), loc(B,[X2,Y2])):-
  X = X2 - X1, Y = Y2 - Y1,
  -0.515502*X <= Y,
  Y <= 0.255538*X,
  pref:(X = -7.2), pref:(Y = 0.0).
  
```

4 まとめ

本論文では、学習システム ACORN を概観し、そこで用いられているインクリメンタルな学習アルゴリズム区間への一般化について説明した。

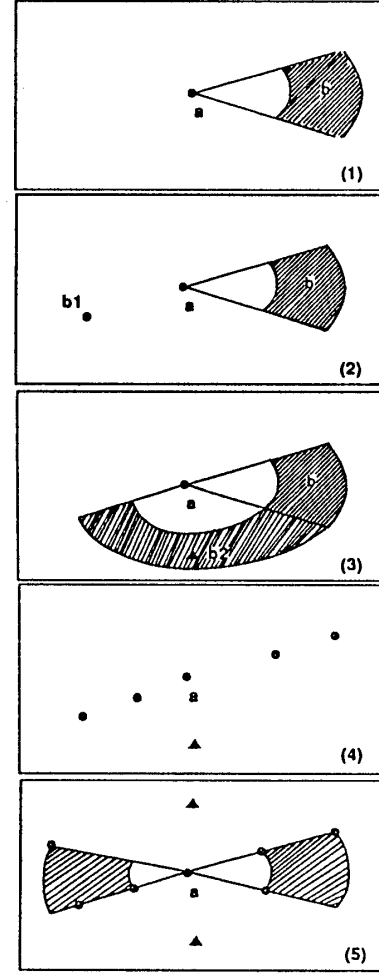


図 3: 空間的関係 *next-to* に対する区間への一般化 (1) (5)

参考文献

- [1] Hiraki, K., Gennari, J., Yamamoto, Y. and Anzai, Y.: Learning Spatial Relations from Images, *Proc. of Eighth International Machine Learning Workshop ML'91*, 407-411, 1991.
- [2] Hiraki, K., Gennari, J., Yamamoto, Y. and Anzai, Y.: Encoding Images into Constraint Expressions, *Proc. of Thirteenth Annual Conference of Cognitive Science Society*, 31-36, 1991.
- [3] 西沢剛, 開一夫, 安西祐一郎: 目標制約の表現機構を備えた制約論理型言語 EPOCH, 人工知能学会論文誌, in press.
- [4] Michalski, R.: A theory and methodology of inductive learning, In Michalski, R., Carbonell, J. and Mitchell, T. (eds.), *Machine Learning: An artificial intelligence approach*. Los Altos, CA: Morgan Kaufmann, 83-134, 1983.