

認知的方略(perceptual-chunks)の学習

—学習プロセスの視覚的制御—

A Perceptual Criterion for Visually Controlling Learning

諏訪 正樹* 元田 浩*
Masaki Suwa Hiroshi Motoda

* (株)日立製作所 基礎研究所
Advanced Research Laboratory, Hitachi Ltd., Hatoyama, Saitama Pref. 350-03, Japan.

1993年5月17日 受理

Keywords: chunking method, diagrammatic reasoning, learning criterion, perceptual-chunks.

Summary

An advantage of diagrammatic reasoning is that it avoids traversing irrelevant paths of inference by controlling search using visual scanning on diagrams. An effective procedure for this in the domain of geometry problem solving is to visually recognize relevant perceptual-chunks in the diagrams of problems, and to use such chunks to guide problem solving. In spite of this beneficial role of perceptual-chunks in diagrammatic reasoning, however, past research has not addressed the issues of acquiring a useful set of perceptual-chunks specific to the target domain, and demonstrating the utility of acquired chunks. This paper addresses these issues by devising a mechanism for learning perceptual-chunks from problem solving episodes. Its basic concept is that the learner acquires, from the problem diagram perceptual chunks each of which is an assembly of diagram elements *that can be visually recognized and grouped together*. Recognition rules implement this chunking criterion in the learning system PCLEAR-N. We show the feasibility of the criterion by presenting experimental data on the operability and cost-effective utility of the learned perceptual chunks in the geometry domain.

1. はじめに

図情報を利用した推論（以後、図情報推論と称する*¹）は近年精力的に研究され始めた新しい研究分野であり、我々人間が図情報を利用して問題解決のための探索を効率的に制御している点が注目を浴びている。では、なぜ図情報推論は効果的なのだろうか？ 過去の研究例の分析により、我々は以下の仮説を得た。それは、「図情報自体が有する二つの性質のおかげで、

我々人間は、問題の図情報を視覚的に走査（visual scan）するだけで問題解決に必要な探索を効率的に制御できる」という仮説である。

第1の性質とは、emergent propertyである*²。「図には物の大きさや空間的位置に関する情報が含まれており、それらの性質（property）に関する推論（例えば比較や抽出）の結果は、図を視覚的に走査するだけで自然に直接浮かび上がってくる」という性質を意味する。例えば、Gelernter[Gelernter 81]は、図情報に含まれる emergent property を後向き推論を枝刈りするために利用している。すなわち、領域知識としてのルールを後向きに適用して生成されるサブゴールが、図情報のなかで正しそうには見えない場合に、そのサブゴールを証明するための推論はそれ以後行わない、というヒューリスティクスとして活用している。また、Koedinger[Koedinger 94]は、図情報を利用す

* 1 英語の technical term は Diagrammatic Reasoning である。問題の Diagram (図情報) を積極的に利用する推論のことを指すが、適切な日本語訳がない。意味的には「図情報を利用した推論」という訳が最も近いが、簡略化して図情報推論と称する。

* 2 Koedinger[Koedinger 92]が命名した性質である。

れば、例えば「角度の等価性」*³などの幾何学的オブジェクトのトポロジカルな性質に関する推論を行わなくてもすむという利点があることを示唆している。

第2の性質とは、locality と compactness である。locality は、「一緒に利用される情報は図のなかではグループ化されて比較的近くに存在する」という性質(文献[Larkin 87]から引用)であり、compactness は「過去に導出されたすべての推論結果が一つの図のなかに表現されている」という性質である。この性質のおかげで、図を視覚的に走査するだけで次に適用可能な知識や必要な情報を効率的に探索することが容易になる。Greeno が提案した physical distance-reducing heuristics[Greeno 83]は、単純に近くに存在するものを次なる探索の対象にするというヒューリスティクスであり、locality という性質をうまく利用したものである。また、Hyperproof[Etchemendy 92]は、ユーザがある命題の成否を判定するよう求められた際に、論理的操作に基づいてその命題を導出するのではなく、図情報から視覚的にその命題の成否を読み取って判定することを許しており、ユーザが図から直接情報を得るための道具として OBSERVE オペレータを用意している。これは、図情報の compactness という性質を利用したものである。まとめると、視覚的な走査を通じて図情報の特質をうまく利用することが、不必要な推論パスを避けるよう探索を制御するという知的問題解決の源である。

視覚的な探索制御の有効な手段として、初等幾何の問題分野に関する研究では次の方法がとられてきた。図のなかにいくつかの「認知的な図形の塊」(以後、perceptual-chunk と称する)を発見し、それらの図形塊を探索のガイドに使うという方法である[Kim 89, Koedinger 90, McDougal 92, Suwa 89a, 諏訪 89b]。perceptual-chunk とは、多くの問題に共通にその一部構造として現れる図形塊であり、認知的に意味ある塊であると考えられる。Kim[Kim 89]は、図情報から対象性または相似性を持つ perceptual-chunk を抽出することの重要性に注目し、「対象性」、「相似性」を表現するための代数的な表記法を提案した。筆者らは、初等幾何の補助線問題においていかにして適切な補助線を発想するかという課題に取り組み、問題の図情報か

ら適切な perceptual-chunk を想起し、それを問題図形に当てはめることにより補助線を生成するという手法を提案した[Suwa 89a, 諏訪 89b]。Koedinger の CD モデル[Koedinger 90]では、問題の図情報をいくつかの perceptual-chunk に分割し、分割に使用された perceptual-chunk に成り立つ性質を順に探索することによって問題解決に要する計算コストが劇的に低減されることを示し、かつこれがエキスパートの問題解決過程をよく説明できることを示唆する認知実験を行っている。McDougal らの POLYA モデル[McDougal 92]は、幾何学的なオブジェクト(点、線分、三角形など)を見る人間の見方を模疑して、視覚的に幾何学的オブジェクトを走査する手続きをインプリメントした初めての試みである。視覚的な走査法は、視覚的行動 (visual action) の連鎖に関する知識(これを search script と呼んでいる)として表現されており、いくつかの search script を活性化した結果として、現問題に関連のある perceptual-chunk を図情報のなかに認識するというモデル化を行った。また、チェスの問題解決においても perceptual-chunk の存在が古くから指摘されてきた[Simon 69]。互いに関連が深いいくつかの駒の位置関係のパターンがそれに相当する。Simon らは、有能なチェスプレーヤーは次の一手を考える準備段階として、版面を視覚的に走査し、現局面をいくつかの perceptual-chunk の重ね合せとして認識することを指摘している。

このように、問題の図情報のなかに perceptual-chunk を発見することは有効な探索制御手段になり得る。それでは、問題領域固有の知識としての perceptual-chunk は、どのようにして獲得すればよいのだろうか? これは重要な研究課題であるにもかかわらず、この分野で研究された例はない。そこで、本論文では、問題を解いた経験から、そのドメイン固有の perceptual-chunk を学習するメカニズムを提案する。このアプローチにおける最重要課題は、問題解決履歴のどの部分を抜き出すかを決定するための学習基準 (learning criterion) として何を与えるかである。説明に基づく学習法 (EBL) の「説明」というコンセプトはそれに対する一つの解決法になっている。つまり、ある目標概念 (target concept) を説明するプロセスに関与した事実だけを「ひと塊」として抽出する。しかし、諏訪らの実験結果[Suwa 94]によれば、EBL は、幾何の問題解決の分野ではあまり適用性の高くない知識を学習してしまう。その本質的な理由は、第1に、幾何の分野における perceptual-chunk とは、視覚的な走査の結果「ひと塊」として認識される図要素であり、

* 3 ある直線上に3点 A, B, C がこの順番に存在し、別の点 D がこの直線上ではない位置に存在するとする。このとき、当然 $\angle DCB$ は $\angle DCA$ に等価であるが、従来の論理的な推論体系でこれを導出するためには、角度の等価性に関する推論ルールを用いるという推論ステップが必要になる。しかし、図情報を利用すれば、その二つの角度が等価であることは自明である。

それは必ずしも EBL により学習できる“説明型知識”が表す図要素に一致しないからであり、第 2 に、幾何の問題解決において支配的な推論形態は perceptual-chunk を視覚的に探索するボトムアップな推論であって、説明型知識を使ってゴールを説明するためのトップダウンな推論ではないからである。

学習基準を与えるという課題に対するもう一つの答えとして、EBL とともに議論された操作性基準 (operationality criterion [Mostow 83]) が考えられる。それは、学習により獲得した知識の操作性が高くなるように学習を制御するための基準である。しかし、操作性基準は「視覚的にひと塊であるものをグループ化する」(以後、視覚的グルーピングと称する) という意味合いを持っていないため、幾何の問題解決において操作性の高い知識を学習することはできない。詳細は 4 章で議論する。

これらの議論から、学習時に我々人間が行う視覚的走査をよく反映し、意味的には“視覚的なグルーピング”を司るような学習基準を与えなければならない。我々は、“recognition rules”という視覚的走査を反映した領域固有知識を新しく提案し、それを学習基準として使用する学習システム PCLEARN をインプリメントした。本論文では、recognition rules がいかに学習プロセスを視覚的に制御するかに焦点を当てて議論する。PCLEARN 法の概念は、過去に筆者らが提案した「フラストレーションに基づく学習法」[諏訪 89b] の拡張により生まれた。「フラストレーションに基づく学習法」が学習の契機に着目し補助線生成用知識を学習する手法であったのに対し、PCLEARN システムは、問題解決中の「オペレータ選択分岐点 (decision point)」を学習の契機にすることによって補助線問題以外の問題領域にも適用できるよう拡張したものである。さらに重要なことに、これは“視覚的グルーピング”という学習概念が付加された結果の産物である。2 章で、recognition rules の基本的な意味合いを述べ、それにより学習を制御する手続きを示す。さらに、我々は、perceptual-chunk を学習するたびにそれを探索制御知識として使用可能な知識のリストに加えて次の問題を解くという実験を行い、学習した知識の操作性および有効性^{*4} を評価した。3 章でその結果を示す。さらに、4 章では、他の学習法と比較し、PCLEARN 法の一般性について議論する。

* 4 一般に、知識のユーティリティと称する [Minton 90]。学習により得た知識を使うことでどれくらいむだな探索が省かれたか、使う際にどれくらいコストを必要とするか、などを定量的に調べた結果として得られる指標である。

2. PCLEARN による学習法

2.1 基本的コンセプト

PCLEARN システムは、ある問題を解いた後に、

- その問題解決履歴中のオペレータ選択分岐ノードのそれぞれを学習の契機とし、
- そのノードそれぞれに関連した「マクロオペレータ情報付きの perceptual-chunk」を問題構造のなかから抽出する。

問題解決中に生成されたあるノードに対して一つ以上の領域ルール (オペレータ) の適用可能性がテストされ、適用が成功したものの一部だけがゴールの証明に寄与し、他の領域ルール (オペレータ) の適用は失敗したかもしくは適用されてもゴールの証明には不必要な適用であった場合に、そのノードをオペレータ選択分岐ノードという。分岐点においてゴールの証明に寄与したオペレータを、以下では、成功オペレータと呼ぶ。学習により獲得した perceptual-chunk は、将来の問題において、それが学習されたときのオペレータ選択分岐ノードと同じ分岐点に遭遇したときに発火し、その分岐点における探索を制御する役割を果たす。

学習プロセスの概念を以下に説明する。

1. おのおののオペレータ選択分岐ノードにおいて、成功オペレータに対応する図要素を「見る」ことが学習の契機になり、
2. そのときに、同時にひと塊として「見える」(本論文では、recognizable という言葉をこの意味で使う) 図要素集合を perceptual-chunk として問題図形から抜き出し、
3. そして、証明木のなかでその perceptual-chunk に対応する部分の IF-THEN 関係を、マクロオペレータ情報として取り出す。

この手続きを実行する際に、ある図要素の「可視性」(「見える」か「見えないか」) を判定する必要が生じる。その判定に使用される知識が recognition rules であり、それが学習における“視覚的グルーピング”を司る中心的役割を果たす。recognition rules に関しては次節で詳説する。注目すべき点は、ある局所的なオペレータ選択分岐ノードではその問題のゴールが必ずしも「見える」とは限らないため、学習された perceptual-chunk が必ずしも問題のゴール構造を含んでいないことである。この点が、あるオペレータの選択が問題の目標概念 (多くの場合、ゴール) を達成するのに貢献した理由を説明する EBL 法との決定的な違いである。

幾何の問題解決分野の代表的な perceptual-chunk の一例が、図4に示されている*5。上段真中の perceptual-chunk のマクロオペレータ情報は、「直線上に並ぶ3点 X, Y, Z に関して $XY = YZ$ というオペレータ選択分岐ノードに遭遇した場合には、点 Y の回りに180度対称な二つの合同な三角形を発見せよ。つまり、そのオペレータ選択分岐ノードで適用すべきオペレータは、「三角形の合同定理である」という探索制御認識を表している。

2・2 Recognition rules

Recognition rules は、意味的には、我々人間がその問題領域のオブジェクトの図表現をどう「見ているか」を知識表現したものである。オブジェクト A が「見える」という命題を *recognizable* (A) と表すと、PCLEARN システムでは、recognition rule を

$$\text{recognizable}(A):- \\ \text{recognizable}(A_1), \dots, \text{recognizable}(A_N), \\ (\text{追加条件式}).$$

という構造として定式化する。つまり、あるオブジェクト (A) が見えるか否かを、そのオブジェクトと part-whole 関係にある他のオブジェクト (A_1 から A_N) の可視性と、「追加条件式」によって規定する。したがって、recognition rule はそれ単独でオブジェクトの可視性を判定するものではなく、初期条件として recognizable なオブジェクト集合が与えられたときにすべての recognizable なオブジェクト集合を網羅するために使用される。「追加条件式」は、その問題領域で使用する述語（領域オブジェクト間に成り立つ性質を表すリテラル）を使って表現する。また、「part-whole 関係」については、以下の例で説明する。

幾何の問題領域における recognition rule を図1に

```
recognizable(X):- recognizable(s(X,Y)).
recognizable(s(X,Y)):- recognizable(a(X,Y,Z)).
recognizable(s(X,Y)):- recognizable(tr(X,Y,Z)).

recognizable(s(X,Y)):-
  recognizable(X), recognizable(Y), exist(s(X,Y)).
recognizable(s(X,Y)):-
  recognizable(X), recognizable(Y), collinear(X,Z,Y).
recognizable(a(X,Y,Z)):-
  recognizable(s(X,Y)), recognizable(s(Y,Z)).
recognizable(tr(X,Y,Z)):-
  recognizable(s(X,Y)), recognizable(s(Y,Z)),
  recognizable(s(Z,X)).

但し、s(X,Y) -- 線分XY、tr(X,Y,Z) -- 三角形XYZ
a(X,Y,Z) -- 角XYZ
下線のリテラルが、追加条件式に相当する。
```

図1 幾何の分野での recognition rule

*5 後で述べるように、図4は PCLEARN が学習した perceptual-chunk の例である。

*6 このプロセスでは、「見える」と判定されたオブジェクトを「見える」オブジェクトの集合に加え、インクリメンタルに可能な限り recognition rule の適用を繰り返す。

示す。この分野のオブジェクトは、点、線分、三角形、角度である。これらのオブジェクトは、互いが他の part-whole 関係にある。part-whole 関係とは、例えば、「点は線分の一部 (part) であり」、「線分は三角形または角度の一部 (part) である」という関係のことである。recognition rule の上記構造は、「あるオブジェクトの「可視性」はそれと part-whole 関係にある他のオブジェクトの「可視性」に依存するであろう」という内省により生まれた。

一般に、あるオブジェクトがすでに「見えている」とときには、そのオブジェクトの一部 (part) であるオブジェクトは、追加条件なしに「見える」と判定できる。例えば、線分 XY がすでに「見えている」とときには、点 X および Y は無条件に「見えている」と判定してよい。図1の上から三つ目までのルールは、この範疇に入る。逆に、複数のオブジェクトから構成される（それらが part であるような）オブジェクトの可視性を証明するときには、おのおのの構成要素のオブジェクトの可視性を証明するだけでなく、さらに構成要素のオブジェクトにある関係が成り立つことを追加条件として証明しなければならないことがある。例えば、線分 XY の可視性を証明するには、構成要素の2点 X, Y の可視性だけでなく、線分 XY が実際に問題図形中に存在する ($exist(s(X, Y))$) か、もしくは別の点 Z があって X, Z, Y 3点が一直線上にある ($collinear(X, Z, Y)$) というどちらかの追加条件が成立しなければならない。図1の上から四つ目以降のルールがこの範疇に属する。ただし、この範疇に入るルールで、追加条件式を必要としないケースも存在する。

2・3 recognition rules による学習制御

問題解決履歴中のオペレータ選択分岐ノードごとに、以下の3ステップの手続きを行い perceptual-chunk を抽出する。

1. ステップ1: 見えるオブジェクトの抽出

オペレータ選択分岐ノードにおいて「見える」オブジェクト集合を以下の手順で列挙する。

- (1) 成功オペレータに含まれるリテラルの引数になっているオブジェクトはすべて recognizable であるとする。これは、成功オペレータの図要素を「見る」ことに相当し、2・1節で述べたように、学習の契機となる。
- (2) (1)のオブジェクトが recognizable であるとしたときに、recognition rule を使用して、recognizable であると判定できるすべてのオブジェクトを列挙する*6。これは、(1)の図要素と一緒に「見

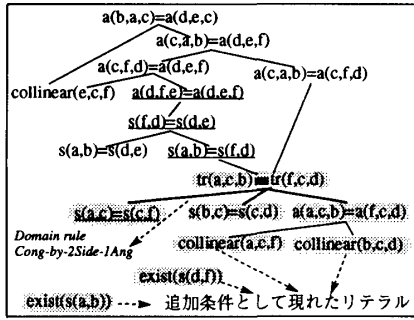


図2 問題14 (付図A) の証明木

える」オブジェクト集合を列挙することを意味する。

図2は付図Aの問題14の証明木を表している。下線を引いたノードがオペレータ選択分岐ノードである。以下では、 $AC=CF$ というオペレータ選択分岐ノードに対して行われる学習手続きを例示する。このノードでの成功オペレータは、「2辺とその間の角が等しい三角形の合同定理」である。この領域知識に登場するオブジェクトは、 $s(b,c)$, $s(a,c)$, $s(c,d)$, $s(f,c)$, $a(b,c,a)$, $a(d,c,f)$, $tr(a,b,c)$, $tr(f,c,d)$ であり、まず、これらのオブジェクトを recognizable とする。これらのオブジェクトの「可視性」から、recognition rules により recognizable であると判定できるオブジェクトをすべて列挙すると、 $a, b, c, d, f, s(a,b), s(d,f), s(b,d), s(a,f), a(b,a,c), a(b,a,f), a(a,b,c), a(a,b,d), a(d,f,c), a(d,f,a), a(f,d,c), a(f,d,b), a(b,c,f), a(a,c,d)$ になる。

2.ステップ：見える性質の導出

次に、ステップ1で得られた「見える」オブジェクトの集合体で成り立つ「見える」性質を抽出する。まず、

- 成功オペレータに登場するリテラル、
- および、ステップ1で recognition rule を使用した際に、あるオブジェクトの可視性を導出するのに必要であった「追加条件式」のリテラル、

を、「見える」性質を導出するための初期条件とする。図2で網がけを施したノードがそれに相当する。このうち、 $collinear(a,c,f)$, $collinear(b,c,d)$, $exist(s(a,b))$, $exist(s(d,f))$ の四つが追加条件として現れたリテラルである*7。次に、これらの初期条件から領域知識を使って導出できるすべてのリテラルを網羅する。そ

*7 $collinear(a,c,f)$ は $s(a,f)$ の「可視性」を判定する際に、 $collinear(b,c,d)$ は $s(b,d)$ の「可視性」を判定する際に現れたリテラルである。

*8 導出に関与しなかったということは、この性質は孤立しており有意な推論に寄与しないものであると判断できるとのことである。

して、初期条件も含めて導出されたすべてのリテラルを「見える」性質とする。

これらの性質を持ったオブジェクト集合を、perceptual-chunk として抽出する。それは、意味的には、オペレータ選択分岐ノードにおいて同時にひと塊に「見える」図要素の集合である。そして、ステップ2で「見える」性質を導出した際に生成された導出木の最下端ノードを IF 部にし、残りのノードを THEN 部にするようなマクロオペレータを、perceptual-chunk のマクロオペレータ情報とする。recognition rule に現れた上述の四つのリテラルのうち、 $exist(s(a,b))$ および $exist(s(d,f))$ は、導出プロセスに関与しなかったため*8、マクロオペレータ情報からは削除される。ここで注意すべきことは、得られたマクロオペレータが、recognition rule 中に現れた追加条件式が追加された分だけ成功オペレータよりも特殊化されているという点である。この特殊性が、抽出された perceptual-chunk の適用性および操作性を左右する。

3.ステップ3：一般化

得られた導出木の各ノードは問題定数に束縛されているので、最終ステップとして、これを領域知識で使われている変数レベルまで一般化する。 $AC=CF$ というオペレータ選択分岐ノードからは、図4の上段真中の perceptual-chunk が抽出される。

以上述べたように、recognition rules は、おのこのオペレータ選択分岐ノードにおいて「見える」性質およびオブジェクトの範囲を決定するための尺度として利用されている。また、「recognizable なオブジェクト集合を初期条件として与えて初めて recognition rules の適用が可能になる」という性質は、recognition rule が問題独立な一般的原則であることを物語っている。つまり、初期条件すなわち学習の契機であるオペレータ選択分岐ノードが異なれば、学習できるチャックも異なる。

3. 実験結果

3.1 学習で得た知識の一般性

学習された知識がその問題分野特有の意味のある perceptual-chunk であるといえるかどうかは、その知識が複数の問題からどれくらい頻繁に学習されるかによる。我々は、20個の初等幾何学の問題(付図Aに示す)を PCLEAR 法および典型的な EBL 法の両方で学習し、学習される知識とその学習頻度を比較するという実験を行った。

EBL システムとして、STRIPS[Fikes 72]のような

表1 perceptual-chunk の学習頻度

Frequency	The number of macro-operators	
	PCLEARN	EBL
1	43	86
2	9	7
3	5	0
4	3	1
more than 4	4	0
total	64	94

初期の EBL システムから, PRODIGY [Minton 89] のように選択的な学習を行うためにメタレベルの目標概念を用意したものまで, さまざまな種類のシステムが提案されている。ここでは, メタレベルの目標概念 SUCCEED に関して学習を行うという PRODIGY の一機能と, PCLEARN 法を比較する。PRODIGY のこの機能は, 問題のゴールノードを目標概念とし, オペータ選択分岐ノードにおいて取ったオペレーの選択がどうして目標概念を達成するのに役立ったのかを説明することによって, 上記の選択分岐ノードでの“preference rule”を学習する。preference rule と PCLEARN 法により学習される知識は, 選択分岐ノードで取るべきオペレータを示唆するという点で同種の探索制御知識なので, この両者を比べることにした。

表1は, 20問の問題から同じマクロオペレータ情報を持つ perceptual-chunk がどれくらい頻繁に学習されたかを, 両方の学習法のそれぞれに関して表したものである。PCLEARN 法が, 数種類のマクロオペレータを複数回学習するのに対して, PRODIGY の学習法は, 違う問題から同じマクロオペレータを学習することはまれであるという結果が示されている。つまり, 初等幾何学の問題領域では, EBL によって学習される知識はもとの問題のゴールに特殊化されがちであり, 異なる問題間に共通に含まれている perceptual-chunk を抽出できない。別の言い方をすると, 幾何学の分野では, EBL のようなゴール構造の学習では意味のある perceptual-chunk を獲得することはできないことを示唆している。

3・2 学習した知識のユーティリティ評価

探索制御知識は, 一般に, 適用可能性をテストされたときに多大のマッチングコストがかかるという負の効果と, オペレータ選択分岐ノードで適用できたときに探索効率を向上させる正の効果を併せ持っており,

* 9 適用を試みたが結局適用できずに終わった場合, 適用できたがそれによって生成されたノードが問題のゴールの証明に寄与しなかった場合, 適用できた生成されたノードが問題のゴールの証明に寄与した場合のすべてを含む。

二つの効果を総合した評価値を知識のユーティリティという [Minton 90]。我々は, PCLEARN 法で学習した perceptual-chunk のユーティリティを評価しながら, ユーティリティの高い知識だけを使って 20 問の問題を解かせ, 問題解決法に現れる学習効果を評価した。

知識のユーティリティは, 以下の式に従って計算する。

$$Utility = TotalSavings - TotalMatchCosts.$$

TotalSavings は, 過去の問題でその知識を使ったことで低減できた計算コストの累計であり, 探索制御知識による re-ordering effect [Minton 90] と呼ばれる効果である。*TotalMatchCosts* は, 過去の問題においてその知識の適用を試みた際にかかったマッチングコスト*⁹の累計である。一つの問題を解くたびに, それまでに学習されているすべての探索制御知識のおおのに関して, re-ordering effect とマッチングコストを計算し, ユーティリティの評価値を更新する。re-ordering effect は以下の方法で計算する。

1. ある知識がオペレータ選択分岐ノードで適用できたとき, それが適用できるまでにそのノードで適用可能な知識を探すのに要したマッチングコスト (コスト 1a と称する), およびその知識の適用により生成された複数の新しいノードで要するマッチングコストの合計 (コスト 1b) の総和を求める。
2. 探索制御知識なしで同じ問題を解いた問題履歴をあらかじめ用意しておく。その問題履歴中で 1 のオペレータ選択分岐ノードに対応するノードを発見し, そのノードからのむだな探索 (もし 1 の探索制御知識を持っていたら行わなかった探索) に要したコスト (コスト 2a), 1 の探索制御知識の適用に関与したノードに対応するすべてのノードでの探索コストの合計 (コスト 2b) を求める。
3. コスト 2 の総和からコスト 1 の総和を引いたものを, re-ordering effect とする。

ある探索制御知識を使うことによってコスト 2a 分だけ得をする正の効果と, 学習により知識の絶対量が増えた結果, おおののノードで適用可能な知識を探索するコストが潜在的に増える負の効果 (コスト 1a + コスト 1b - コスト 2b の分だけ増加) のトレードオフで, この知識のユーティリティが決定される。つまり, 個々の知識のユーティリティ評価値は他の知識のマッチングコストに依存することになり, 全体の知識量の増加による探索マッチングコストの増大が, 個々の知識のユーティリティ評価値を全体的に押し下げる結果となる。したがって, 知識がたまるにつれて, 正のユーティリティ評価値を得られるような優良な知識は, 適用

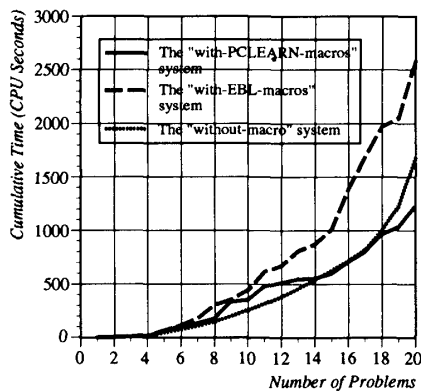


図3 学習曲線 (PCLEARN 法と EBL 法の比較)

したときにコスト 2a が非常に大きいようなものだけに限られる。

実験方法を示す。20 問の問題を解いていく過程で、1 問解くたびに学習を行うと同時にそれまで保有している探索制御知識のユーティリティを計算し、次の問題を解くときにはユーティリティが上位の一定個*10の知識だけを使用可能とした。与える問題の順番は、学習なしに解いたときに計算コストの小さい順とした。

図3は、解いた問題の数を横軸にし、それまでに解いたすべての問題の問題解決コストの累積を縦軸にして、以下の三つのケースの学習曲線を示したものである。PCLEARN 法で学習しながら解いた場合、EBL 法で学習しながら解いた場合、学習なしで解いた場合である。

最初の数問を経過した時点では、PCLEARN システムは学習なしの場合に比べて多くのコストを要しているが、これは知識量の増加に伴う探索マッチングコストの増加によるものである。しかし、11 問目を過ぎるあたりから、学習曲線の傾きが学習なしの場合に比べて緩やかになる。つまり、学習された知識の適用が生む re-ordering effect が現れ始めているといえる。一方、EBL システムの学習曲線は、傾き自体も学習なしの場合と比べてさほど変わらず、累積コストは学習なしの場合よりも多くかかっている。学習で得た知識があまり適用されることがなく、マッチングコストばかりかかっていることを表している。

表2~表4のデータは、EBLで獲得した探索制御知識が幾何の分野において学習効果を示さないのはなぜかを物語っている。表2は、PCLEARN法とEBL法に関して、20問の問題を解く間に、オペレータ選択分岐ノードにおいて、獲得した知識の適用をテストするのにかかった全コストを示している。1回のテストに

*10 本実験では25個に設定した。

表2 学習された知識の適用テストに要するコスト

	Total (msec)	Frequency of testing	Cost per one testing (msec)
PCLEARN	288,733	1,146	252
EBL	976,963	1,910	511

表3 学習された知識の適用性

	Frequency of testing	Frequency of applying	Percentage (%)
PCLEARN	1,146	39	3.4
EBL	1,910	12	0.6

表4 学習された知識のサイズ分布

	All macros		Applied macros	
	Average	Standard Deviation	Average	Standard Deviation
PCLEARN	3.3	1.4	2.7	1.3
EBL	5.0	1.8	3.2	1.2

要する平均コストは、EBL法で学習した知識(以後、EBL知識と称する)はPCLEARN法で学習した知識(以後、PCLEARN知識と称する)の約2倍である。表3は、20問の問題を解く間に、学習された知識が適用された延べ回数を示している。PCLEARN知識の適用確率(オペレータ選択分岐ノードでテストされた総回数に対して、適用が成功した回数の比率)は、EBL知識の適用確率に比べ、約6倍ほど高いことが観察できる。EBL知識の学習効果の低さは、そもそも適用確率が低いことに起因しているといえる。では、なぜ、EBL知識はマッチングコストが大きく(表2)、適用確率が低い(表3)のだろうか? 表4は、PCLEARN知識およびEBL知識のそれぞれに関して、適用できた知識のサイズの分布(平均および標準偏差値)を、学習により獲得されたすべての知識のサイズの分布(平均および標準偏差)と比較したものである。ここで、知識のサイズとは、簡単のため、そのIF部の条件の個数とする。PCLEARN知識の場合、適用できた知識のサイズの分布は、すべての知識の分布とあまり変わらない。これは、PCLEARN法が、適用性の観点から見て適切なサイズの知識を生成していることを示唆している。それに対し、EBL知識の場合、学習されたすべての知識の分布は、適用できた知識の分布に比べて、著しくサイズが大きい側に偏っていることが観察できる。これは、すなわち、EBL知識は適用性の観点から見てサイズが大きすぎることを意味している。

以上の結果をまとめると、EBL法は、オペレータ選択分岐ノードから問題のゴールに至る推論パスをチャンクするという原則で学習するため、学習知識のサイズが大きくなりすぎる傾向があり、それが原因で学習された知識は多大のマッチングコストを要するものになる。また、上記の原則のために、学習知識はもとの

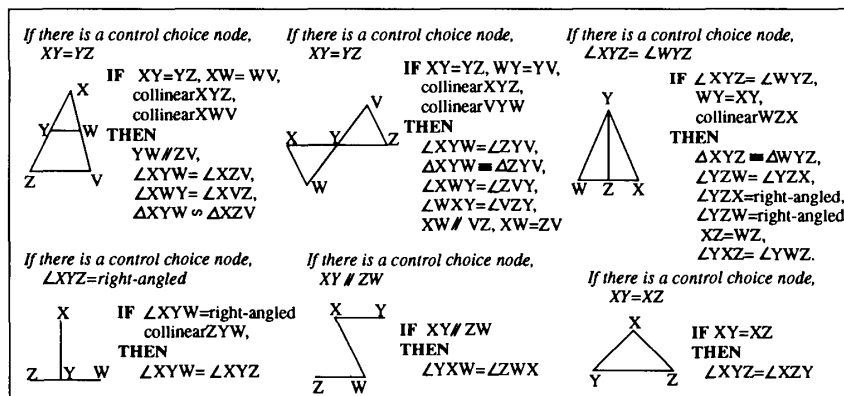


図4 PCLEARN 法により学習された perceptual-chunk (正のユーティリティ評価値を得たもの)

問題のゴール構造を反映して特殊化されすぎたものになり、適用性が低くなる。つまり、適用性自体が低く、適用されたとしてもマッチングコストが大きいが、結局、EBL 知識の学習効果を低下させている原因であると結論づけられる。一方、幾何の分野での実験によれば、PCLEARN 法の recognition rules は、学習知識の大きさを制御するための学習尺度として、効果的である。つまり、適用性の面で適切なサイズを持ち、優れた cost-effective 性能を発揮する探索制御知識を学習できることが示された。図4は、おのおのの知識のユーティリティを評価しながら20問の問題をすべて解いた結果として、最終的に正のユーティリティ評価値を得た perceptual-chunk の集合を示している。

4. 考 察

4.1 Operability Criterion との違い

PCLEARN 法の学習尺度と EBL システムの operability criterion は、根本的に異なることを本節で議論する。両者は、チャンキングの目的、意味合い、手法のすべてに関して異なるコンセプトに基づいている。

PCLEARN 法のチャンキングの目的は、Koedinger の CD モデル[Koedinger 90]に見られるように、問題空間をいくつかの部分空間に分割しむだな推論パスを生成しないようにするための知識を学習することにある。一方、EBL 法において operability criterion を設ける目的は、将来の問題(例えば認識問題)において高い適用性を発揮するような言葉で記述された知識を学習することである。

両者の目的の違いは、学習の意味合いの違いを生む。PCLEARN 法によるチャンキングは、人間がその領域のオブジェクトを眺めたときに“ひと塊に見えるもの”をチャンクとして抽出するという意味合いを持ってい

る。それに対し、operability criterion によるチャンキングは、“適用性のある”(operational)言葉(認識問題の場合、“機能的な”(functional)言葉ではなく、“叙述的な”(descriptive)言葉)だけで記述された塊を問題の説明木から抽出するという意味合いを持つ。ここで、“叙述的な”言葉で表現されたものは“ひと塊に見える”ものに等しいのではないか? したがって、その点で PCLEARN 法の学習尺度は operability criterion の拡張に過ぎないのではないかと考える読者がいるかもしれない。しかし、それは誤りである。むしろ、operability criterion は、問題空間をいくつかの部分部分に分割して眺めることに使える知識を獲得するための“視覚的なグルーピング”というコンセプトをいっさい含んでいないことから、両者はまったく異なる学習法だと考えるのが自然である。operability criterion の概念を説明するのにしばしば使われる次の例題を考えてみよう。“飲むためのカップ”の概念は、機能的な言葉(例えば、「手で持てる」という言葉)を領域知識を使って叙述的な言葉(例えば、「とってががついている」という言葉)に変換することにより学習できる、という例題[Winston 83]である。operability criterion の観点からすれば、機能的な言葉と叙述的な言葉の区別は重要であるが、PCLEARN 法の観点からすれば、大事なものは表現する言葉の性質の区別ではなく、人間が目を見たときにひと塊にグループ化されるものとそうでないものとの区別である。カップは非常に簡単な構造をしているがゆえに、人間がそれを複数の部分構造に分割して眺めるというような対象物ではないことが、両学習法の違いをわかりにくくしているかもしれない。しかし、とても複雑で人間ならそれを複数の部分構造に分割して眺めようとするような対象物があったとして、その対象物に関して学習を行う例題を考えよう。その対象物の概念を説明する説明木は、機能的あるいは叙述的な

多くの言葉で表現されたノードからなるであろう。この対象物に関して PCLEARN 法が行う学習は、“視覚的なグルーピング”という観点から対象物を複数の部分構造に分割しその要素の重ね合せとして対象物を認識することであり、対象物を叙述的な言葉だけで表現した説明知識を抽出することではない。この例題では、両学習方法は明らかに異なる学習結果を示すと考えられる。

学習の目的および意味合いを実現するために、両学習法はまったく異なる手法をとる。EBL 法では、目標のタスクにおいて適用性が高いとわかっている叙述的な言葉のリストをあらかじめ用意しておき、それらの言葉で目標概念が表現されるように説明プロセスを導き、概念を学習する。PCLEARN 法では、おのおののオペレータ選分岐ノードで視覚的にひと塊に見えるものをチャンクするという性質上、チャンクされるものはオペレータ選択分岐ノードに依存し、あらかじめそれを表現するための言葉を用意することは不可能である。したがって、「人間がある領域のオブジェクトのイメージを想起する際に他のどのようなオブジェクトと一緒に想起するか」に関する知識だけを与えておき、おのおのの選択分岐ノードにおいてその知識に照らして“ひと塊に見える”ものを決定するという手法をとる。

4・2 他の学習法との比較

PCLEARN 法を, Soar [Laird 87] および ACT 理論 [Anderson 83] と比較する。Soar および ACT 理論は、問題解決器が生成したサブゴールを満足することに関与したオペレータ列を学習器がチャンクして、探索制御知識とするという学習法である。我々は、EBL 法も含めて、これらの学習法を“ゴール指向型チャンキング”と呼ぶ。その理由は、これらの学習法は、いずれも、問題空間中のサブゴールや目標概念が学習の契機となり、それらを直接達成するためのゴール指向的なマクロ知識を学習するという共通のコンセプトに基づいているからである。これに対して、PCLEARN 法が学習する知識はゴール指向的ではない。問題解決器のゴール指向的な問題解決行動のひとまとまりをチャンクするのではなく、問題解決中の人間の視覚的な認識に照らしてひとまとまりのオペレータ列をチャンクするため、獲得される知識は、サブゴールや目標概念を達成するためのものではなく、問題解決中のある局面でひと塊の推論パスを想起し、局所的にその局面を打開するための知識になる。

4・3 PCLEARN 法の一般性

初等幾何の問題解決においては、PCLEARN 法の有効性が示された。ここでは、この学習法が適用できる問題のクラスについて議論する。

PCLEARN 法は、問題の図情報をいくつかの部分に分割して認識するための知識を学習する手法で、それによって、問題解決器は、問題解決計画をたてる際に重要な推論ステップだけに注目できるようになる。この種の視覚的な認識に基づく問題解決計画は、特に、図情報推論の分野で研究されてきたドメインで重要である。初等幾何の問題解決 [Koedinger 90, McDougal 92, Suwa 89], 空間的に配置された構造体における挙動予測 [Narayanan 91, Narayanan 93], 空間的情報を有する積木世界での推論 [Etchemendy 92] などである。

学習を支える最も重要なプロセスは、問題の図情報を視覚的に走査することであり、本学習法では、recognition rules の使用でそれを模擬的に実現している。したがって、本学習法が上記の問題領域で適用可能かどうかは、それぞれの問題領域で recognition rules を記述できるかどうかにかかっている。recognition rules は、人間がその問題領域のそれぞれのオブジェクトが「見える」ための条件を記述したものである。recognition rules は、part-whole 関係にある領域オブジェクト相互の「可視性」の関係を想起するだけで問題解決プロセスとは独立に記述することができ、それにもかかわらず学習時には、ひと塊に認識できる図情報要素を決定する視覚的な学習尺度として機能する。

5. 結 論

問題の履歴からドメイン特有の perceptual-chunk を学習する手法 PCLEARN 法を提案した。PCLEARN 法は、問題解決履歴中の各オペレータ選択分岐ノードにおいて、ひと塊に見える図要素部分を抽出して perceptual-chunk とするという手法をとり、EBL のように問題解決履歴から目標概念を達成するためのゴール指向的な知識を学習する手法とは根本的に異なる。その新規性は、図要素を抽出するための学習尺度として、互いに part-whole 関係にある領域オブジェクトの「可視性」の関係を記述した知識 recognition rule を提案したことにある。このルールは、学習に際して人間が図情報を視覚的に走査する方法を意味表現したものであり、このような視覚的な基準を学習プロセスの制御に利用した初めての試みである。

学習で獲得した perceptual-chunk を探索制御知識

として使うことは、意味的には、問題の図情報を複数の部分図要素の重ね合せとして認識することによって、むだな推論パスをなくすべく問題解決プロセスを視覚的に制御することを意味する。ただし、PCLEARN システムの現在のバージョンは、適用可能な perceptual-chunk を視覚的なマッチングで探すのではなく、論理表現された記述のマッチングにより探すという手法に頼っている点が、人間本来のメカニズムとは異なる。

初等幾何の問題解決の分野において、学習された

perceptual-chunk の探索制御知識としての有効性を評価するための実験を行い、PCLEARN 法は、EBL に比べて、適用性が高くユーティリティの優れた探索制御知識を学習できることを示した。

本研究は、図情報推論に必須の perceptual-chunk を自動獲得するものであるが、これは、図情報的な知識および論理的に表現された知識の両方を、学習や問題解決プロセスにおいて利用するハイブリッド推論システムを実現するための第1段階であると理解してほしい。

◇ 参 考 文 献 ◇

- [Anderson 83] Anderson, J. R.: *The Architecture of Cognition*, Harvard University Press, London, England (1983).
- [Etchemendy 92] Etchemendy, J. and Barwise, J.: Hyperproof: Logical Reasoning with Diagrams, *Working Notes of AAAI Stanford Spring Symp. on Diagrammatic Reasoning*, pp. 80-84 (1992).
- [Fikes 72] Fikes, R. E., and Hart, P. E. and Nilsson, N. J.: Learning and Executing Generalized Robot Plans, *Artif. Intell.*, Vol. 3, pp. 251-288 (1972).
- [Gelernter 91] Gelernter, H.: Realization of a Geometry-Theorem Proving Machine, E. A. Feigenbaum and J. Feldman (eds.), *Computers and Thought*, pp. 134-15, R. E. Krieger Publishing Company Inc. Reprint Edition (1981).
- [Greeno 83] Greeno, J. G.: Forms of understanding in mathematical problem-solving, *Learning and Motivation in the Classroom*, Erlbaum (1983).
- [Kim 89] Kim, M. Y.: Visual Reasoning in Geometry Theorem Proving, *Proc. IJCAI-89*, pp. 1617-1622 (1989).
- [Koedinger 90] Koedinger, K. R. and Anderson, J. R.: Abstract Planning and Perceptual Chunks: Elements of Expertise in Geometry, *Cognitive Science*, Vol. 14, p. 511-550 (1990).
- [Koedinger 92] Koedinger, K. R.: Emergent Properties and Structural Constraints: Advantages of Diagrammatic Representations for Reasoning and Learning, *Working Notes of AAAI Stanford Spring Symp. on Diagrammatic Reasoning*, pp. 154-159 (1992).
- [Laird 87] Laird, J. E. and Newell, A. and Rosenbloom, P. S.: SOAR: An Architecture for General Intelligence, *Artif. Antell.*, Vol. 33, No. 1, pp. 1-64 (1987).
- [Larkin 87] Larkin, J. and Simon, H. A.: Why a diagram is worth ten thousand words., *Cognitive Science*, Vol. 11, pp. 65-99 (1987).
- [McDougal 92] McDougal T. and Hammond, K.: A recognition model of geometry theorem-proving, *Proc. 14th Annual Conference of the Cognitive Science Society*, pp. 106-111 (1992).
- [Minton 89] Minton, S., Carbonell, J. G., Knoblock, C. A. Kuokka, D. R., Etzioni, O. and Gil, Y.: Explanation-Based Learning: A Problem Solving Perspective, *Artif. Intell.*, Vol. 40, pp. 63-118 (1989).
- [Minton 90] Minton, S.: Quantitative Results Concerning the Utility of Explanation-Based Learning, *Artif. Intell.*, Vol. 42, pp. 363-391 (1990).
- [Mostow 83] Mostow, D. J.: Machine Transformation of Advice into a Heuristic Search Procedure, *Machine Learning: An Artificial Intelligence Approach*, CA: Tioga Press (1983).
- [Narayanan 91] Narayanan, N. H.: Reasoning Visually about Spatial Interactions, *Proc. IJCAI-91*, pp. 360-365 (1991).
- [Narayanan 93] Narayanan, N. H., 諏訪正樹, 元田 浩: A Case for Depictive Representations, 人工知能学会基礎論研究会, SIG-FAI-9301, pp. 1-8 (1993).
- [Simon 69] Simon, H. A. and Barenfeld, M.: Information-processing Analysis of Perceptual Processes in Problem Solving, *Psychological Review*, Vol. 76, No. 5, pp. 473-483 (1969).
- [Suwa 89a] Suwa, M. and Motoda, H.: Acquisition of Associative Knowledge by the Frustration-Based Learning Method in an Auxiliary-line Problem, *Knowledge Acquisition*, Vol. 1, pp. 113-137 (1989).
- [諏訪 89b] 諏訪正樹, 元田 浩: 初等幾何学におけるフラストレーションに基づく学習, 人工知能学会誌, Vol. 4, No. 3, pp. 308-320 (1989).
- [Suwa 94] Suwa, M. and Motoda, C.: Learning perceptually-chunked macro operators, *Proc. Int. workshop on Machine Intelligence* (also to appear in *Machine Intelligence volume 13*, Oxford University Press) (1994).
- [Winston 83] Winston, P. H., Binford, T. O., Katz, B. and Lowry, M.: Learning Physical Descriptions from Functional Definitions, Examples and Precedents, *Proc. of AAAI-83*, pp. 433-439 (1983).

[担当編集委員・査読者: 安西祐一郎]

◇ 付 録 ◇

1. **Givens:** ADC collinear, $\angle ADB = \text{right-angled}$, $\angle ABD = \angle CBD$. **Goal:** $AD = CD$

2. **Givens:** BDC collinear, $AB = AC$, $\angle BAD = \angle CAD$. **Goal:** $BD = CD$

3. **Givens:** $AB = AC$, $AE \parallel BC$, BAF collinear, $\angle FAE = \angle CAE$. **Goal:** $BD = CE$

4. **Givens:** $AB = AC$, $\angle ABD = \angle CBD$, $\angle ACE = \angle BCE$, AEB collinear, ADC collinear. **Goal:** $BD = CE$

5. **Givens:** AEC collinear, BED collinear, $AB = AD$, $BC = DC$. **Goal:** $\angle BEA = \text{right-angled}$

6. **Givens:** BDHC collinear, $DH = HC$, $\angle DHA = \text{right-angled}$, $\angle DBA = \angle DAB$. **Goal:** $BD = AC$

7. **Givens:** BEC collinear, AFE collinear, $AD \parallel BC$, $\angle ABF = \angle EBF$, $\angle BAE = \angle DAE$. **Goal:** $\angle BFA = \text{right-angled}$

8. **Givens:** AFC collinear, BGD collinear, $AF = FC$, $BG = GD$, $AB = CD$, $\angle AFE = \text{right-angled}$, $\angle BGE = \text{right-angled}$. **Goal:** $\triangle ABE \cong \triangle CDE$

9. **Givens:** $BD = DC$, $AF = AC$, $DE \parallel BF$, BAF collinear, BDC collinear, CEF collinear. **Goal:** $\angle AEC = \text{right-angled}$

10. **Givens:** $AB \parallel CD$, $AD \parallel BC$, $BM = MC$, ABP collinear, PMD collinear, BMC collinear. **Goal:** $AB = BP$

11. **Givens:** $AG = GD$, $BG = GE$, $CG = GF$, AGD collinear, BGE collinear, CGF collinear. **Goal:** $\angle ABC = \angle DEF$

12. **Givens:** ADB collinear, CED collinear, $AC = BC$, $AE = BE$. **Goal:** $\angle ADC = \text{right-angled}$

13. **Givens:** AEB collinear, ADC collinear, BDM collinear, CEN collinear, $AE = EB$, $AD = DC$, $BD = DM$, $CE = EN$. **Goal:** $AN = AM$

14. **Givens:** $BC = CD$, $AC = CF$, $AB = DE$, BCD collinear, ACF collinear. **Goal:** $\angle BAC = \angle DEC$

15. **Givens:** ADC collinear, AEF collinear, BED collinear, BFGC collinear, $AD = DC$, $BE = ED$, $AF \parallel DG$. **Goal:** $BF = CG$

16. **Givens:** BNM collinear, AEN collinear, ADM collinear, $\angle ABD = \angle CBD$, $\angle ACE = \angle BCE$, $\angle ADB = \text{right-angled}$, $\angle AEC = \text{right-angled}$. **Goal:** $DE \parallel MN$

17. **Givens:** ADB collinear, ACE collinear, DME collinear, BFM collinear, $BD = CE$, $CM = MF$, $DM = ME$. **Goal:** $AB = AC$

18. **Givens:** RPQ collinear, APB collinear, CAQ collinear, BRC collinear, $AB = AC$, $\angle BRQ = \text{right-angled}$. **Goal:** $AP = AQ$

19. **Givens:** AEC collinear, ADB collinear, DYE collinear, BXC collinear, AYX collinear, $AD = DB$, $AE = EC$. **Goal:** $AY = YX$

20. **Givens:** BAR collinear, BPC collinear, APE collinear, AQDC collinear, PQR collinear, $AB = CD$, $BP = PC$, $AP = PE$, $AQ = QD$. **Goal:** $AQ = AR$

付図 A 実験に使用した初等幾何学の問題

著者紹介

諏訪 正樹 (正会員) は, 前掲 (Vol. 9, No. 2, p. 200) 参照.

元田 浩 (正会員) は, 前掲 (Vol. 9, No. 1, p. 16) 参照.