小特集 「帰納論理プログラミング」

帰納論理プログラミングの応用
－近年の国際会議の動向を踏まえて－

Applications of Inductive Logic Programming – New Trends

沼尾 正行* Masayuki Numao

* 東京工業大学大学院情報理工学研究科
Graduate School of Information Science and Engineering, Tokyo Institute of Technology.

1997年7月25日 受理

Keywords: machine learning, inductive logic programming, information extraction, design, music.

1. はじめに

本小特集最初の論文 [古川 97] にもあるように、帰納論理プログラミングの分野では、しっかりした基...

2. 自然言語処理

ICML ’97 ではテキサス大学オースティン校のRaymond Mooney が招待講演を行い、ILP の自然言語処...

Sep. 1997

帰納論理プログラミングの応用 683
Telecommunications. SOLARIS Systems Administrator. 38-44K. Immediate need

Leading telecommunications firm in need of an energetic individual to fill the following position in the Atlanta office:

SOLARIS SYSTEMS ADMINISTRATOR
Salary: 38-44K with full benefits
Location: Atlanta Georgia, no relocation assistance provided

埋められたテンプレート:

computer_science_job
title: SOLARIS System Administrator
salary: 38-44K
state: Georgia
city: Atlanta
platform: SOLARIS
area: telecommunications

図1 文書とそれに対応するテンプレートの例

Extraction Rules）と呼ばれる学習システムを開発した。
RAPIER は 図1 に示すような文書およびスロットを
埋められたテンプレートのペアを訓練例として使って,
スロットを埋める情報のパターンおよびその周囲のパ
ターンを学習する。

学習されたルールを用いれば、情報として欲しい項目
を前処理や後処理なしで、文書から直接抽出できる。
たとえば、図1 に基づいて、分野(area)を抽出する
ためのルールを生成すると、以下のようになる。

<table>
<thead>
<tr>
<th>直前のパターン</th>
<th>1) 語: leading</th>
</tr>
</thead>
<tbody>
<tr>
<td>スロットを埋める</td>
<td>1) リスト: 長さ: 2</td>
</tr>
<tr>
<td>情報のパターン</td>
<td>タグ: [nn,nns]</td>
</tr>
<tr>
<td>直後のパターン</td>
<td>1) 語: [firm,company]</td>
</tr>
</tbody>
</table>

このルールは leading と firm, leading と company に
よって囲まれる 2 語以下の単語が分割を示す句（図1の
場合, telecommunications）であることを示している。

「タグ:」の中ににある nn, nns といったタグは POS
(Part-Of-Speech) タグ付けプログラム [Brill 94] によ
るものである。

帰納論理プログラミングのための一般化の手法とし
て最小汎化 (Least General Generalization) [古川 92]

がある。これはもちろん論理プログラムを対象としたも
のであるが、それを上述のやや特殊な規則形式に適用で
きるように改造して、学習アルゴリズムとして用いる。
詳細については原典で吟味頂くとして、ここでは一般
化の方法を例に沿って説明する。「located in Atlanta, Georgia.」および「offices in Kansas City, Missouri.」
という二つの句から、両者が満たす最小限の一般化（す
なわち, 最小汎化）を生成することを考える。

located in Atlanta, Georgia. に対応するルール:

<table>
<thead>
<tr>
<th>直前</th>
<th>1) 語=located タグ=vbn</th>
</tr>
</thead>
<tbody>
<tr>
<td>2) 語=in タグ=in</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>スロット</th>
<th>1) 語=Atlanta タグ=nnp</th>
</tr>
</thead>
<tbody>
<tr>
<td>直後</td>
<td>1) 語=, タグ=,</td>
</tr>
<tr>
<td>2) 語=Georgia タグ=nnp</td>
<td></td>
</tr>
<tr>
<td>3) 語=, タグ=,</td>
<td></td>
</tr>
</tbody>
</table>

offices in Kansas City, Missouri. に対応するルール:

<table>
<thead>
<tr>
<th>直前</th>
<th>1) 語=offices タグ=nns</th>
</tr>
</thead>
<tbody>
<tr>
<td>2) 語=in タグ=in</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>スロット</th>
<th>1) 語=Kansas タグ=nnp</th>
</tr>
</thead>
<tbody>
<tr>
<td>2) 語=City タグ=nnp</td>
<td></td>
</tr>
<tr>
<td>直後</td>
<td>1) 語=, タグ=,</td>
</tr>
<tr>
<td>2) 語=Missouri タグ=nnp</td>
<td></td>
</tr>
<tr>
<td>3) 語=, タグ=,</td>
<td></td>
</tr>
</tbody>
</table>

一般化の手順として、まず直前と直後のパターンを空
にし、スロットの情報を埋めるパターンとして 2 種類の
一般化を生成する。Kansas City は 2 語なのに、Missou-
riu は 1 語なので、[2 語以下の語] に一般化する。語
の制約は全語の選択が制約なしのどちらかに一般化さ
れる。全項目でのタグの制約が同じならば最小汎化の
タグの制約も同じになる。Atlanta, Kansas, City の
3 語は辞書中の単一の意味クラスに属さないので、意味
は制約されない。一般化された規則は次のようになる。

<table>
<thead>
<tr>
<th>直前</th>
<th>空</th>
</tr>
</thead>
<tbody>
<tr>
<td>スロット</td>
<td>1) リスト</td>
</tr>
<tr>
<td>長さ=2 語=[Atlanta, Kansas, City]</td>
<td></td>
</tr>
<tr>
<td>タグ=nnp</td>
<td></td>
</tr>
<tr>
<td>直後</td>
<td>空</td>
</tr>
</tbody>
</table>
これらのルールは誤った例を満たすので、直前および直後のパターンの最小を追加する。 “in” の項目と “,” の項目が上の二つの規則で同一条なので、そのまま用いる。辞書が州に対して意味を含むと仮定して州の名前を一般化すると、「州の名前という意味制約、npn というタグの制約、語の制約なし」というパターンで、「州の名前という意味制約、npn というタグの制約、二つの州名と選言」というパターンが生成される。最終的には字が最も多くなる正例を満たすルールとして出力される。

<table>
<thead>
<tr>
<th>直前</th>
<th>空</th>
</tr>
</thead>
<tbody>
<tr>
<td>スロット</td>
<td>1) リスト 長さ=2 語タグ=nnp</td>
</tr>
<tr>
<td>直後</td>
<td>空</td>
</tr>
</tbody>
</table>

論理プログラムの場合とは異なり、最小を追加する過程はかなりアドホックな感じがあるが、これは探索空間を制限するために、現時点ではやむを得ないであろう。90 個の訓練例から作成された規則の精度は平均 83.7% となっており、実用にも十分耐えられると考えられる。

本例題の学習結果は経済学論理式に比べると単純である。しかし、それを決定するよりは考えてみると、文中の数個の組み合わせについて、それが例えば City スロットに適するかどうかを判定する決定木を構築せねばならず、現実的とはいかない。その上、属性の種類も、「[直前-1]-語」「[直前-1]-タグ」「[直前-2]-長さ」といった具合に組み合わせ的に増えてしまい、属性の値域には、数十語の中から数語を取り出す組み合わせ（例えば, [Atlanta, Kansas, City]）なども含まれ、膨大になる。このことは、ニューラルネットを用いても同じであり、ILP の技法を応用する価値は大きいと言えよう。

### 3. ソフトウェア工学

ILP を C++ のクラスの誤り予測に利用する研究が行われている [Cohen 97]。この問題は、ソフトウェア工学で注目されている。メリーランド大学でデータが収集されている。被験者の学生は、中規模の情報管理システムの開発を要求され、その結果を独立したソフトウェアの専門家グループがテストして、誤を含んだ部分を記録する。

<table>
<thead>
<tr>
<th>直前</th>
<th>空</th>
</tr>
</thead>
<tbody>
<tr>
<td>スロット</td>
<td>1) 語=1 タグ=in</td>
</tr>
<tr>
<td>直後</td>
<td>1) 語=1 タグ=nnp</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>直前</th>
<th>空</th>
</tr>
</thead>
<tbody>
<tr>
<td>スロット</td>
<td>1) リスト 長さ=2 語タグ=nnp</td>
</tr>
<tr>
<td>直後</td>
<td>空</td>
</tr>
</tbody>
</table>

この問題では、得られるデータが少ないため、種々の背景知識に頼らざるを得ない。また、プログラム中の構造を最大限に利用して予測を行うと意味で、ILP に適した問題であると考えられる。Cohen らは、Prolog を数の制約を記述できるように拡張し、その学習システム FLIPPER を構築して、上の問題について実験を行った。その結果、FOIL および C4.5 などに比べて、よい結果を得ることができたと報告している。

### 4. 音楽感官情報処理

コンピュータにより、音楽情報を獲得する研究では、ニューラルネットワーク [Todd 91]、もっとも実用的な説明に基づく学習 [沼尾 94] などがあり用いられてきた。しかしながら、楽譜とは音符、その組合せおよび和音、和音のシーケンスなどの複雑な入れ子構造があることを考えると、ILP に適した問題の一つであると考えられる。

このような観点から、山口等は自動編曲システムへの ILP の応用を試みた [山口 93]。このシステムは、図 2 に示すように、ILP の一種である理論洗練システム [沼尾 94] を用いている。理論洗練システムには各領域で成立する領域理論（＝Prolog のプログラム）が与えられ、例題を用いてその間違いを修正（＝プログラムをデバッグ）することにより、正しい領域理論を得ることができる。領域理論として、音楽教科書に書いてある和音の理論を用いると、メロディより得られたコードの列を基に編曲を行うことができるが、その結果は標準的な編曲結果であるとは言え、いささか味気無いものになる。

音楽を理解するには、知性というよりは「感性」 [辻 95] であろう。感性は完全なパターン、あるいは標準的なパターンから生じるのではなく、それからの「差異」や「ずれ」の部分によって生じるのだという考え、
方がある [石原 95]. 作曲家は標準からわずかずれた編曲を行うことにより、「感性」を醸し出す編曲を行うのである. このような曲をあらかじめ、組み込むことで、このような曲をとくことはできないし、できたとしても個性に依存するのであるから、すぐに変更する必要が生じるであろう. さらに、習作によって「まね」するのに最適の対象である. このシステムは、作曲家の編曲を支援として理論指導を行い、そのような「まね」を獲得する. 少々ブローキングな結果結果は少ないが、かなり面白い結果が得られ、被験者には好評であった.

このシステムをさらに発展させ、特定の人が感性を獲得したり、感性に合わせた編曲を可能にしたのが [Noma 97] である. 編曲によって得られる音符や和音などの構造のどの部分が人に感性を抱かせるのかを獲得するのが目標で、ごく部分的にはあるが、そのような構造を取り出すことを成功している. また、そのような構造が人によって微妙に異なることも確認している. 以上的結果より、示明した理論を獲得するだけではなく、漠然とした対象を捉えるために ILP が有用であることが確かめられたことになる.

5. 三次元形状の設計および二次元レイアウトの認識

設計は、数多くのトレードオフを選択しながら、目標を満たすデザインを求める問題であり、主に個人的な経験に依拠し、機械学習の技術を応用できる場合面は多い. 形状設計、構造設計、回路設計などいろいろな分野があるが、どの場合においても細部が部分形状同士の関係を取り扱う必要があり、ILP の手法が有用である.

たとえば、プラスチック製品の形状は自由に決定できるように思えるが、ちょっとした違いで金型の複雑さが大きく変化し、コストが高くなる. プラスチック製品が安価なのは、一つの型で繰り返し同じ形状の製品を製造するからであり、型に合成樹脂を流し込み、型を用いて製品を取り出すために、その形状に制約がある. 図 3 (a) はコストの安い形状であり、上下に二つの型を設け、型を用いて製品を取り出すのは容易である. (b) の場合は、上下二つの型では一型が製品の突起に引っかかってしまって、抜けができない (アンダーカット). この容器は上部が若干開き、大きめになっているので、突起のところを型を分割すると、今度は上部の型が抜けなくなる. 左右二つの型とした場合には、底面の穴および側面に型が引っかかってしまう. 結局、底面に穴をあけ、側面を形成するための三つの型を用いて、スライド金型を用意するしかなかった.

実際の設計では、金型については素人の設計者が (b) のような形状を設計し、専門家がコストダウンのために、(a) に修正するという過程を繰り返している. そのような設計過程より、(b) を負例、(a) を正例として, CAD システム自身が学習を行えば、コストの低い形状を自動的に提示するような設計システムを構築できる. それを目指して学習の実験が行われた [守屋 95]. 製品の形状は、部分形状の組合せとして、図 4 のような関係で表現できる. これらの関係を組み合わせて、ILP により学習させた. 160 個の設計例を用いて実験を行っ
紙面のレイアウトの認識を可能にするものである。紙面のレイアウトの高次情報の関係で表現するのが適切である。ILP を用いると上のベンチマークでは 96%以上の正解率が得られており、他の手法より優れていることが実証された [Zhang 97]。

近年、OCR 技術の発達により、文字単位では 100%近く認識が行えるようになっているが、紙面のレイアウトを認識できないため、いざ実際の書類を認識させてみると、意外に認識率が低くなってしまう場合が多い。OCR ソフトウェアを大局から制御する技術として、以上のようなレイアウトの認識手法が有効であろう。

6. 教育支援システム

ILP の起源とも言えるシステムとして、MIS [Shapiro 82] があるが、80 年代にこれを知的チューブに応用する研究が盛んに行われた。ユーザのパーセンタがコンピュータ側で学習することにより、ユーザモデルを自動構築しようというわけである。しかしながら、MIS は極めて効率が悪いため、次第に顧みられなくなった。

近年、ILP 技術の革新により状況は大きく変わりつつあり、実際に動作するシステムが紙面と発表されている [Baffes 96, Sison 97]。また、現在の ILP システムは、従来語として新しい述語をユーザに質問せずに効率的に発明する機能を備えている [Kjisirikul 92a, キッスリクル 92b]。これを用いると、ユーザモデルを構築する際に不足する語彙（述語）を自動的に発明することが可能になる [若月 95]。

7. おわりに — 今後の課題

帰納論理プログラミングの応用について最近動向を紹介した。ILP の主にヨーロッパにおける応用研究のサーベイ [Dzeroski 96] にもあり、本解説では紹介しかなかった研究が数多く収録されている。興味のある方は是非参照して頂きたい。論理学習の研究と異なり、ILP では変数が導入でき、それが実用問題における探索空間の拡大を招き、応用を難しくしている。無数の変数の導入を抑える手法 [Mofizur 96] も提案されているので、応用の各ケースにおいて評価を行っていくことが必要である。

ILP の研究を応用しやすくなると、感性情報や設計の場では、もう少しあいまいな情報を扱える柔軟性が必要である。そのような研究も徐々に出始まっており、ルールの重み付けたり、述語論理と同様に表現力をもった重み付けネットワークを用いることなど

*3 http://www.ics.uci.edu/~mlearn/
が提案されている [Numao 97]。また、ニューラルネットワークに階層論理を扱わせる研究 [Botta 97] も行われており、その過程で、バックプロパゲーションのようなアルゴリズムと ILP との関係も明らかになっていくであろう。また、あいまいな情報を扱うのは若干異なるが、例題にノイズが入っている場合や、確率的な探索を行う方法、連続値を扱う手法なども研究されている [Lavrač 96]。

謝辞

ILP による自然言語処理について研究している修士2年の野崎宏明君をはじめ、いろいろと議論をして頂いた研究室の学生に、この場を借りて礼を述べたい。

参考文献


[辻 95] 辻信三郎. 感性情報処理の情報学——心理学的考察. 平成4年度～平成5年度科学振興費補助金 (重点領域研究) 研究成果報告書 (課題番号222), 1995.


著者紹介

沼尾 正行（正会員）は、前掲 [Vol.12, No.4, p.590] 参照、http://numao-www.cs.titech.ac.jp/ (numao@cs.titech.ac.jp)