帰納論理プログラミングを用いた語義判別規則の学習

○阿部修也　　新納浩幸

茨城大学 工学部 システム工学科

1 はじめに

本論文では，帰納論理プログラミング（Inductive Logic Programming，ILP）を利用して，語義判別規則の学習を試みる．

自然言語処理の個々の問題を，分類問題として定式化し，帰納学習の手法により解決するというアプローチは大きな成功をおさめている．そして，そこで利用される帰納学習手法は，主に確率統計的な手法である．しかし分類問題はILPを利用しても解くことができる．ILPは確率統計的な手法にはない特徴も有しており，問題によっては実用的な手法として利用されている[5][7].

確率統計的な手法にはないILPの大きな特徴は，背景知識の利用である[9]．背景知識は，訓練データとして与えられた情報とは別の情報のことを指す．この背景知識をうまく利用することで，確率統計的な手法よりも精度の高い規則を学習できる可能性がある．

本論文ではSENSEVAL2の日本語TMタスク[6]における語義判別問題にILPの手法を適用した．ILPの実装システムとしてはProgolを利用した．背景知識としては分類語彙表を利用した．実験の結果，ILPが確率統計的な手法と同等以上の規則を学習できた．また分類語彙表を用いることでより精度の高い規則を学習できた．

2 Progolによる分類問題の解法

ILPは，Muggleton[1][2]，Quinlan[3]等により開発された体系である．ILPの実装では，Muggletonによる Progol[2] や Quinlan による Foil[4] が有名である．本実験では，Progolを使用する．

Progolは以下の特徴を持つ．

- 背景知識を利用し，帰納推論に基づいて学習を行う．
- 一階述語論理に基づく，高い表現能力をもつ．

Progolが分類問題に対する分類規則を学習できることを示すため，ある動物が哺乳類なのか，それ以外の類なのかを判別する分類問題を考える[10]．表1では，授乳するという特徴が，哺乳類のみに共通する特徴である．これは，哺乳類だけが授乳し，それ以外の類（魚類，鳥類）は授乳しないということを示している．つまり，哺乳類であることを分類する規則は，「授乳」である．

表1: 哺乳類による分類問題の例

動物名	類	授乳	足の数	住処	恒温
犬	哺乳類	○	4	陸上	○
イルカ	哺乳類	○	0	水中	○
鮫	魚類	×	0	水中	×
鷲	鳥類	×	2	空中	○

Progolは，事例（動物名）とその素性（授乳，足の数，住処，産卵，恒温）から，クラス（哺乳類）を判別する規則を学習する．

Progolは，事実を述語論理で表現するため，事例も素性もクラスも述語論理で記述する．例えば，犬に関しては次の様に記述する．

```
class('犬', yes).          % 犬は哺乳類
milk('犬', yes).           % 犬は授乳される
legs('犬', 4).             % 犬の足の数は4本
habitat('犬', '陸上').     % 犬の住処は陸上
homothermic('犬', yes). % 犬は恒温動物
```

Progolの入力ファイルとProgolによる実行結果を，図1に示す．正例は学習対象となる事例とそのクラスを表し，背景知識は事例の素性，それ以外はバイアスとしてProgolの動作を制御する．バイアスの中でも，modehは学習対象を，modebは学習対象の素性をProgolに伝える．

実行結果の，`class(A,yes) :- milk(A,yes).`は，“Aが授乳するならば，クラス（類）が哺乳類”（Aは変数）ということを示している．哺乳類とそれ以外の類を分類する規則は「授乳」であることを，正しく学習することができている．

```
% %以降，行末まではコメント

% 正例のみの学習
:- set(posonly)?

% モード宣言
:- modeh(*,class(+animal, #yn))?
:- modeb(*,milk(+animal, #yn))?
:- modeb(*,legs(+animal, #nat))?
:- modeb(*,habitat(+animal, #habitat))?
:- modeb(*,eggs(+animal, #yn))?
:- modeb(*,homeothermic(+animal, #yn))?

% タイプ宣言
animal('犬'). animal('イルカ').
animal('鮫'). animal('鷲').
yn(yes). yn(no).
habitat('陸上'). habitat('水中').
habitat('空中').

% 正例
class('犬', yes). class('イルカ', yes).
class('鮫', no). class('鷲', no).

% 背景知識
milk('犬', yes). legs('犬', 4).
habitat('犬', '陸上'). eggs('犬', no).
homeothermic('犬', yes).

milk('イルカ', yes). legs('イルカ', 0).
habitat('イルカ', '水中'). eggs('イルカ', no).
homeothermic('イルカ', yes).

milk('鮫', no). legs('鮫', 0).
habitat('鮫', '水中'). eggs('鮫', yes).
homeothermic('鮫', no).

milk('鷲', no). legs('鷲', 2).
habitat('鷲', '空中'). eggs('鷲', yes).
homeothermic('鷲', yes).
```

```
% Progol による学習結果は次のようになる.
class(A,yes) :- milk(A,yes).
class(A,no) :- milk(A,no).
```

図 1: 哺乳類に共通の規則の学習

3 語義判別問題への応用

語義判別問題は分類問題として定式化できる．Progol は分類問題を解くことができるため，語義判別問題を解くことができる．語義のクラスは Progol では正例にあたり，文章の素性が背景知識に相当する．

素性として，語義判別の対象となる単語の前後にある単語を用いる．具体的には，語義となる単語の直前の単語（“e1”）と直後の単語（“e2”），それを除いた，前方最大 3 単語（“e3”），後方最大 3 単語（“e4”）を使用する．

判別する語義が「与える」の例を示す．与えられた文章は，「彼では力不足という印象を与えるかもしれない」．この文章を単語分割すると「彼」「で」「は」「力」「不足」「と」「いう」「印象」「を」「与える」「かも」「しれ」「ない」と分割される．この文章の素性は次のようになる．

- 直前の単語 e1 = 'を'
- 直後の単語 e2 = 'かも'
- 前方の単語 e3 = {'と', 'いう', '印象'}
- 後方の単語 e4 = {'しれ', 'ない'}

Progol の入力形式で表わすと図 2 のようになる．Progol では，素性は背景知識として扱われる．また，ここで文章の ID とは，文章を識別するための番号である．

```
% 正例
class('文章の ID', 'クラス名').

% 背景知識
e1('文章の ID', 'を').
e2('文章の ID', 'かも').
e3('文章の ID', 'と').
e3('文章の ID', 'いう').
e3('文章の ID', '印象').
e4('文章の ID', 'しれ').
e4('文章の ID', 'ない').
```

図 2: 素性を Progol の入力形式にした場合

4 背景知識の利用

例えば，「宿題」と「課題」という単語を考える．この 2 つの単語は異なる単語だが，同じ意味を持っていると考えられる．しかし，異なる単語が同じ意味を持つ可能性をシステムは知らない．語義判別問題は，単語の意味に基いてルールを生成した方がよい．そのため異なる単語でも意味が同じならば，同じ単語として扱った方がよい．

上記の考えを背景知識として組み入れるために，分類語彙表を利用する．分類語彙表は，階層構造を持つ．階層が上がるほど，同じ意味の単語は増加する．（図 3）．

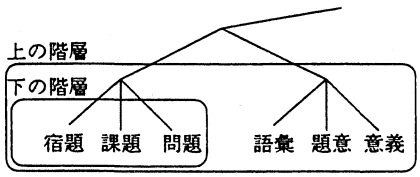

図 3: 分類語彙表の一部

一番下の階層を考えると，「課題」「宿題」「問題」は，同じ意味のグループである．「語彙」「題意」「意義」は，別の意味のグループである．同じ意味のグループに属する単語は，同じ意味を持つ．1 つ上の階層を考えると，「課題」「宿題」「問題」「語彙」「題意」「意義」という単語は同じ意味になる．

規則（図 4）は，素性（e1, e2, e3, e4 の単語）と，分類語彙表に含まれる単語と，その単語と同じ意味を持つ単語を結び付ける．つまり，Progol は同じ意味の単語を同じ単語として扱うようになる．述

語 b は，分類語彙表の単語とその意味を表わす数字の組を表す．述語 e1 ～ e4 は，文章の素性を表す．述語 e1_w ～ e4_w は，述語 e1 ～ e4 の代りとして，文章の素性を表す．また，規則の記述を簡略化するために，素性が分類語彙表に存在しない場合は，あらかじめ素性を分類語彙表に登録する．

```
e1_w(ID, Word1) :- e1(ID, Word2),
                   b(Word2, Number), b(Word1, Number).
e2_w(ID, Word1) :- e2(ID, Word2),
                   b(Word2, Number), b(Word1, Number).
e3_w(ID, Word1) :- e3(ID, Word2),
                   b(Word2, Number), b(Word1, Number).
e4_w(ID, Word1) :- e4(ID, Word2),
                   b(Word2, Number), b(Word1, Number).
```

図 4: 素性と分類語彙表を結ぶ規則

5 実験

SENSEVAL2

SENSEVAL2[6] は語義判別のコンテスト形式の国際会議であり，2001 年 7 月に ACL-01 に併設されて開催された．SENSEVAL2 の日本語タスクには辞書タスクと翻訳タスクがあるが，本実験で扱うのは翻訳タスクである．翻訳タスクでは TM と呼ばれる日英対訳データが予め与えられる．そして対象単語を含むテスト文が出題され，その対象単語を英訳する際に利用できる TM の例文番号を解答として返すタスクである．例文番号をクラスと考えれば，翻訳タスクは分類問題として定式化できる．また TM の例文が正例と背景知識の基となる．

規則の生成

TM の例文を，JUMAN（形態素解析）を用いて単語分割する．単語分割が，失敗または適切ではない部分は，手作業で修正する．ここで，単語の一部を素性（e1, e2, e3, e4）として抽出する．それとは別に，例文番号をクラスと考え，例文ごとにクラス付けを行う．例文番号，クラス，素性を述語に変換し，Progol の入力ファイルを生成する．入力ファイルを Progol に読み込ませて，規則を生成する．

解答の作成

テストは 40 単語が対象である．各単語ごとに 30 問，計 1200 問のテストとなる．それらに対して，Progol により生成された規則を使い，語義判別を行った．

実験結果

分類語彙表を使用しない場合の正解率は 48.67 %．分類語彙表を使用すると正解率は 54.00 % に向上した．分類語彙表を使用しない場合の正解率は，SENSEVAL2 において TM だけを用いて学習したシステムと，ほぼ同等の正解率である [6]．分類語彙表を使用すると正解率は 5.33 % 増加し，同等以上の正解率を得られた．背景知識が有効に利用され，より優れた規則が生成されたと言える．

Progol と確率的手法の 1 つである決定リストを直接比較してみる．論文 [8] では TM の例文だけを訓練データとして決定リストを作成し，54.00 % の精度を出している．ただしそこでは，例文をグループ化して，クラスの数を減らしている．実際に TM タスクでは例文番号をクラスとして扱うよりも，例文をグループ化して，同じグループの例文には同一のクラスを与えた方が精度が高くなる．

ここでも論文 [8] で利用したクラスと同一のクラスを用いて，Progol により学習させた結果，60.50 % の結果を得た．ただし，ここでは分類語彙表を用いていない．同一クラスで学習した場合でも，確率的手法に比べて高い正解率となった．これは，ILP が背景知識を用いずとも，十分に優れた規則を生成することができるためである．

6 考察

ILP と確率的手法

Progol と決定リストによる結果を比較した．結果は Progol の方が優れていた．しかし，確率的手法には決定リストよりも優れた結果を出すアルゴリズムが存在する．そのため，ILP を使った手法が明かに優れているとは言えない．ただし今回の実験のように訓練データが少ない場合には，訓練データにない情報，つまり背景知識の利用方法の優劣が重要になってくる．そのような問題には ILP が有利になると思われる．

背景知識の悪影響

論文 [8] と同一のクラスを用いて，今度は分類語彙表を使用したところ，正解率が 60.50 % から 58.92 % へと 1.58 % 悪くなった．背景知識を用いることで，生成される規則が悪くなっているためである．しかし，大きく正解率が下っているわけではない．

この実験で利用したクラスは，ほぼ最適にグルーピングした結果得られたものであることと，分類語彙表を使用する前の高い正解率を考慮すると，分類語彙表の使用前でも，十分に優れた規則が生成されていたと考えられる．そのため，分類語彙表がノイズとなり，規則生成に悪い影響を与えたと考えられる．

通常，背景知識は有益な情報である．しかし，システムがこれを有効に扱えなければ，背景知識はノイズで結果に悪影響を与えてしまう．背景知識を有効に利用し，悪影響を最小限に防ぐことが今後の課題である．

規則の優先順位について

Progol が生成する規則にはいくつか特徴がある．

- デフォルト規則は 1 つあればよいにもかかわらず，複数生成されることがある．
- デフォルト規則が生成されないことがある．
- 規則は入力ファイルで述語が記述された順番に上から生成され，生成された順番のまま出力される．そのため，生成された規則には優先順位がない．

全ての事例に適用することができる規則をデフォルト規則と言い，その他のどのルールにも適用されない事例に適用される．

通常，規則に優先順位が付けらているか，優先順位で並べられていた方がよい．しかし，Progol には規則の優先順位がないため，どの規則が優先されるべき規則なのかがわからない．特に，デフォルト規則は 1 つしか用いられないにもかかわらず，デフォルト規則が複数生成される．そのため，どのデフォルト規則を用いるべきかが大きな問題になる．（図 5）

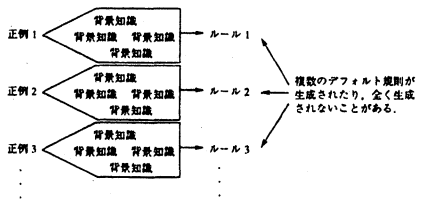

図 5: デフォルト規則の生成

今後，確率的手法を取り入れた PRISM，SLP といった実装システムを用いることで，これらの問題に対応したい [10].

7 終りに

本論文では ILP の実装システムとして Progol を利用し，SENSEVAL2 の日本語 TM タスクにおける語義判別問題に ILP の手法を適用した．背景知識としては分類語彙表を利用した．実験の結果，ILP が確率統計的な手法と同等以上の精度の規則を学習できた．また分類語彙表を用いることでより精度の高い規則を学習できた．

しかし，背景知識を有効に利用することができない場合もあった．今回は条件によって背景知識が悪影響をおよぼしたが，今後はさまざま条件で背景知識を有効に利用できる方法を探りたい．また，確率的手法を取り入れた PRISM，SLP といった実装システムを用いることで，規則生成の優先順位に関する問題に対応したい．

参考文献

[1] Muggleton, S.: Inductive logic programming, *New Generation Computing*, 8, pp.295-318 (1991).

[2] Muggleton, S.: Inverse Entailment and Progol, *New Generation Computing*, 13, pp.245-286 (1995).

[3] Quinlan, J. R.: Learning Logical Definitions from Relations, *Machine Learning*, 5, pp.239-266 (1990).

[4] Quinlan, J. R. and Camereon-Jones, R. M.: Induction of Logic Programs: FOIL and Related Systems, *New Generation Computing*, Vol.13, pp.287-312 (1995).

[5] Srinivasan, A., Muggleton, S. H., King, R. D. and Sternberg, M. J. E.(1994) Mutagenesis: ILP experiments in a non-determinate biological domain. In Proceedings of the Fourth International Workshop on Inductive Logic Programming.

[6] 黒橋禎夫，白井清昭：SENSEVAL-2 日本語タスク，信学技報，Vol101, No.351, pp1-8 (2001).

[7] 嶋津恵子，古川康一：データベースからの知識発見システム DM-Amp – 設計と実装とエキスパートシステム開発への応用，人工知能学会誌，Vol.15, No.4, 論文特集：「発見科学」，pp.629-637.

[8] 新納浩幸：SENSEVAL-2 日本語翻訳タスクに向けて作成した語義判別学習システム Ibaraki，信学技報，Vol101, No.351, pp25-30 (2001).

[9] 古川康一：帰納論理プログラミング — チュートリアル—，人工知能学会誌，Vol.12, No.5, 小特集：「帰納論理プログラミング」，pp655-664 (1997).

[10] 古川康一，尾崎知伸，植野研：帰納論理プログラミング，共立出版（2001）