# 帰納学習システムの比較検討と応用可能性*

7P－6

溝口文雄　大和田勇人　清水健一†
東京理科大学　理工学部‡

## 1　はじめに

知識ベースの整備が進み，取り扱うデータベースが巨大化した現在，効率良く有用な知識を獲得することが重要な課題となっている．そして，近年の帰納論理プログラミングの研究の進展により，様々な帰納学習システムが開発され，複雑な現実問題へと適用可能となってきている．学習システムはボトムアップ／トップダウン型に大別でき，効率性の観点から両者の代表を見ると，前者がGOLEM[3]，後者がProgol[4]である．

そこで本稿ではGOLEM，Progolを対象とし，代表的な論理プログラムの例題やドラッグデザインなどのデータに対する適用の実験を通じて，両システムを比較するとともに，大量データを扱うような現実問題への適用について検討を行なう．そして問題点の解消方法を提案する．

## 2　学習システムの一般化手法

事例から帰納的に学習を行なうシステムが様々に開発されている．一般に帰納学習は，正事例集合$\mathcal{E}^+$，負事例集合$\mathcal{E}^-$，背景知識$\mathcal{K}$を用いて$\mathcal{K} \cup \mathcal{H} \vdash \mathcal{E}^+$かつ$\mathcal{K} \cup \mathcal{H} \not\vdash \mathcal{E}^-$を満たす仮説$\mathcal{H}$を導出する．帰納学習のアプローチには，最も一般的な仮説を特殊化していくトップダウン的なものと，最も特殊な仮説を一般化していくボトムアップ的なものがある．しかし，従来のシステムでは膨大な仮説空間を探索しなくてはならないという問題があり，効率良く帰納学習を行なうためには，仮説空間と探索空間の縮小が課題となる．

仮説空間を縮小するために，GOLEMは以下のような制約を用い効率良く帰納的一般化を行なう．

- 事実は基底項に限る
- ヘッドに現れる変数は必ずボディにも現れ，ボディに現れる変数は一意に定まらなくてはならない(決定性の概念)
- ボディに現れるリテラルの数の制限($ij$-determination)

GOLEMは，ボトムアップのアプローチで，ある事例のペアについて最も一般化した仮説を生成し，包含する正事例を増やし負事例を増やさない方向で仮説を一般化していく．

また，探索空間を縮小するために，Progolは次の戦略により一般化を行なう．

- 精密化子による節の枚挙
- ヒューリスティックな評価関数を用いた最良優先探索

Progolはトップダウンのアプローチで，１つの事例について最も特殊化した節を特殊化の上限とし，単位節を包含する負事例を減らす方向で一般化していく．

GOLEMでは，事実の記述が基底項に限られる，記号領域しか学習できないなどの問題点もある．Progolはこの点を解消し，確定節，実数領域も学習できるという利点を持つ．

*Examination and Applicability of Inductive Learning Systems
†Fumio Mizoguchi, Hayato Ohwada, Kenichi Shimizu
‡Faculty of Sci. and Tech., Science Univ. of Tokyo

## 3　代表的な論理プログラムの例題への適用

animals, member, mult, qsortの問題に対して，GOLEM, Progolを適用した結果を表1に示す．なお，multの学習の例においてProgolでは，plus/3などの背景知識を`plus(0,0,0)`, `plus(0,1,1)`のように基底項で与える場合と，`plus(X,Y,Z):-Z is X+Y.`のように確定節の形式で記述した場合について，別個に実行した(GOLEMでは基底項でしか事実を記述できない)．

表1: GOLEM，Progolの各データに対する実行結果

| GOLEMの実行結果 | animals | member | mult 基底項 | mult 確定節 | qsort |
|---|---|---|---|---|---|
| 定義した節の数 | 122 | 20 | 278 | - | 84 |
| 正事例の数 | 16 | 15 | 34 | - | 16 |
| 負事例の数 | 45 | 3 | 6 | - | 4 |
| 実行時間(ms) | 1086 | 972 | 10062 | - | 13348 |
| 学習規則数 | 4 | 2 | 1 | - | 4 |
| 残された事例数 | 0 | 0 | 4 | - | 0 |

| Progolの実行結果 | animals | member | mult 基底項 | mult 確定節 | qsort |
|---|---|---|---|---|---|
| 定義した節の数 | 121 | 18 | 242 | 9 | 19 |
| 正事例の数 | 16 | 16 | 36 | 5 | 11 |
| 負事例の数 | 42 | 3 | 7 | 21 | 12 |
| 実行時間(ms) | 448 | 192 | 30182 | 285 | 2131 |
| 学習規則数 | 4 | 2 | 3 | 2 | 2 |
| 残された事例数 | 1 | 0 | 0 | 0 | 0 |

定義した節の数とは，正事例，背景知識と(モード宣言に関する)タイプの節の数の総和である．実行時間は，Sunワークステーション上で５回ずつ実行させた平均時間である．

表1からProgolがGOLEMから実行速度の点で改良されていることが分かる．ただし，全て基底項で記述したmultの学習の例においては，ほぼ同じ入力内容にもかかわらず，Progolの実行時間はGOLEMの3倍程度となっている．これは定義した節の数が多い場合，Progolのアルゴリズムでは探索空間が膨大となってしまうためである．

## 4　実際の問題への適用

GOLEMは，ドラッグデザイン[2]，タンパク質二次構造予測，有限要素メッシュデザインなど様々な現実問題に対し適用されてきた．ここでは，ドラッグデザイン(正事例数941，背景知識数581)のデータに対しGOLEM，Progolを適用した．学習された規則により予測されたランクと本来のランクとの相関係数を表1に示す．GOLEMの結果が良いのは，一般化規則を生成し獲得する際，手動作によって良さそうな規則を選んでいったためである．

```
great(A,B):-struc(B,C,D,h),struc(A,E,F,G),
    hb_donnor(E,h_don0),polar(E,H),great0_polar(H),
    less5_polar(H),flexibility(E,I),less2_flex(I).
```

GOLEMにより得られた上記のような一般化規則はProgolでは得られにくく，Progolでは次のような規則が主に得られた．

表 2: ドラッグデザインのデータへの適用結果

| | GOLEM | Progol |
|---|---|---|
| 本来／予測ランク間の相関係数 | 0.879 | 0.591 |

```
great(A,B):-struc(A,C,D,D),struc(B,E,E,D).
great(A,B):-struc(A,C,D,C),struc(B,E,C,C),polar(E,2),
    flexibility(D,F),F=<1.
```

その理由は，Progol はその探索の性質上，ボディ部のリテラルの数がより少ない規則を選びやすいということと，得られる規則が 1 つの事例の性質を強く引き継ぐことにある．例えば背景知識において `struc(name,h,h,h)` とあれば，`struc(A,C,C,C)` とリテラルを生成してしまう．これらのことから，GOLEM の方が Progol よりも，より精度の良い規則を導出すると考えられる．

## 5 一般化手法の問題点の比較検討

3,4節の実験結果から，GOLEM，Progol の大量データを扱う問題への適用の際の問題点を考察する．

### 5.1 GOLEM に関して

実行毎に結果が異なるため，望ましい解を得るためには，手動作により繰り返し実行させる必要がある．データが基底項，記号に限られるため，学習可能領域が限定され，また，定義しなければならない事実の数が多くなる．

学習される規則の精度は良いと考えられるが，事例数が数万以上の大規模データベースを扱うには，実行途中でシステムエラーを起こしてしまうことがあることなどから，システムの信頼性は高くなく，その実行能力は充分とはいえない．

### 5.2 Progol に関して

定義節の数が多い場合と探索深さが深い場合，探索ノード数は膨大なものとなる．出力値が変数／定数であるかまで厳密にモード宣言しなくてはならず，どちらの結果も必要な時は二重に定義し，その結果探索空間は倍増してしまう．また，GOLEM とは異なり，一括処理を行ない全ての事例を説明するまで結果を出力しないため，事例数があまりにも多い場合探索，Progol は百時間以上でも実行を続けるが規則を得るのが困難となる．

一般化を行なう際，1 つの事例に対する特殊化から始めるので，学習される規則は採択した事例の性質に大きく依存し，事例の入力順序に大きく影響を受ける．mult/3 の学習の例でも，事例の入力順序によって正しい解が得られないことがある．このことから，事実を線形式で表せるような整合性のあるデータに対しては適しているが，ノイズを含むようなデータに対してはあまり適していないとも考えられる．

### 5.3 GOLEM，Progol の性能比較のまとめ

GOLEM，Progol の性能比較のまとめを表 3に示す．

## 6 効率的な特殊化手法の提案

5.2節では Progol の重要な問題点として次を挙げた．

- 定義してある節の数が多くなれば，仮説の候補が膨大になり，探索空間の縮小が図れない
- 採択した事例の性質に一般化規則は依存する

この問題点を解消するために，ボディにリテラルを付加して仮説を特殊化していく際に次のヒューリスティクスを導入する．

- ボディに付加するリテラルの候補の中で，入力から決定される事例の数が最も含まれるリテラルを付加していく．

例えば mult/3 の学習の例において，

表 3: GOLEM，Progol の性能の比較

| | GOLEM | Progol |
|---|---|---|
| 探索方向 | ボトムアップ | トップダウン |
| データ型 | 記号 | 記号・数値 |
| データ形式 | 基底項 | 基底項・確定節 |
| 処理方式 | 手動作 | 一括処理 |
| 探索解 | ランダム | 一定 |
| 実行速度 定義節数多 | 中 | 低 |
| 定義節数小 | 中 | 高 |
| システム信頼性 | 低 | 高 |

$\mathcal{E}^+$:{`mult(0,1,0),mult(1,1,1),mult(1,2,2)`}

$\mathcal{K}$:{`plus(0,1,1),plus(1,1,2),dec(1,0),dec(2,1)`}

とすると，`mult(A,B,C)` の深さ 1 での一般化のボディのリテラルの候補は，{`mult(A,A,D)`, `mult(B,B,E)`, `plus(A,A,F)`, `plus(A,B,G)`, `plus(B,B,H)`,`dec(A,I)`,`dec(B,J)` } であるが，このとき $\mathcal{E}^+$,$\mathcal{K}$から出力変数が全て具体化できるのは `dec(B,J)` だけなのでこのリテラルをボディに付加する．このような操作を繰り返していくことで特殊化を進める．

この特殊化におけるヒューリスティクスの導入により，効率的に解に到達できるだけでなく，1 つの事例からではなく事例全体から一般化が進められるので，ノイズを含むような現実問題への適用に適していると考えられる．

この特殊化手法を関係データベースの結合演算を導入 [1] することにより実現した学習システムを，4節のドラッグデザインのデータに適用したところ，予測したランクと本来のランクとの相関係数 0.649 を得た．この値は Progol の結果よりも良く，特殊化の手法の改良により精度の良い規則が導出されたといえる．

## 7 まとめ

本稿では，GOLEM，Progol の性能比較，及び，現実問題への応用の可能性について検討した．Progol は GOLEM などの従来の帰納学習システムと比較して，学習可能領域の拡大と推論の高速化を実現した．しかし，大量データを扱う現実問題の適用に際しては幾つかの問題点があり，また，その探索方法には効率化の余地が見込まれることが実験を通じて考察された．

Progol の特殊化を改良した一般化の手法では，事例全体から一般化を進めており，1 つの事例からの一般化よりもノイズを含むような現実問題への適用に適していると考えられる．この一般化手法による学習システムの実装，及び，現実の問題領域への応用を実験・検証することが今後の課題である．

## 参考文献


[1] 石井雅子，大和田勇人，溝口文雄: 帰納論理プログラミングに基づく関係データベースからの規則の導出，日本ソフトウエア科学会第 11 回大会論文集,pp.333-336,1994.

[2] R.King,S.Muggleton:Drug Design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase , *Proc. of the National Academy of Science*,89(23),pp.11322-11326,1992.

[3] S.Muggleton,C.Feng: Efficient Induction of Logic Programs, *Proc. of the Workshop on Algorithmic Learning Theory*,1990.

[4] S.Muggleton: Mode-Directed Inverse Resolution, *Machine Intelligence 14*,Oxford University Press(to appear).