3E-09

帰納論理プログラミングにおける統計値に基づいた仮説探索の効率化

黒田 洋介　　大原 剛三　　馬場口 登　　北橋 忠宏

大阪大学 産業科学研究所

1 はじめに

近年，帰納論理プログラミング (Inductive Logic Programming: **ILP**) による知識獲得が盛んに研究されている．ILP による知識獲得は，目標概念の正例，負例，及び背景知識が与えられた時に，1つの負例も導くことなく，全ての正例を導くような仮説を獲得するものであり，その獲得方法は仮説候補からなる探索空間において最適な仮説を見つけだす探索問題と捉えることができる [1]．しかし，実世界のデータベースへの適用を考えた場合，探索空間内の仮説数が爆発し，探索に膨大な時間がかかる傾向がある．そこで，本研究では，Inverse Entailment(**IE**)[2] を用いたトップダウン探索に対して，対象データの統計的解析結果に基づき探索空間を縮小する手法を提案する．

尚，本研究ではデータベースへの ILP の適用を前提とし，データベース中に記述される各事例に関する属性と属性値の組を背景知識として用いるものとする．

2 IE を用いた仮説探索

IE を用いた仮説探索では，背景知識と1つの正例 e から，正例 e を導く仮説の中で最も特殊な仮説 (Most Specific Clause: **MSC**) を生成する．次に MSC をボトム，恒偽をトップとする一般・特殊関係を半順序とした仮説候補束内を，トップからボトムに向けて最も評価値の高い仮説 (最適仮説) を見つける為に探索する．仮説の評価関数には，主に仮説が導く正例数，負例数に基づいたものが用いられ，その探索空間の大きさは変数への代入の組合せを考えなかった場合，MSC の条件部のリテラル数 n に対して，$\sum_{i=1}^{n} {}_nC_i = 2^n$ となる．

例えば図 1に示すような正例，負例，背景知識が与えられた場合，正例 $bird(owl)$ に IE を適用すると以下のような MSC が得られる．

$$bird(owl) \leftarrow fly(owl, true), \\ cover(owl, feather), egg(owl, true). \quad (1)$$

ここで記号「←」は含意記号であり，リテラル $fly(owl, true)$ とは事例 owlは属性 flyにおいて属性値 $true$ をとるという意味である．

正例：	bird(owl)	bird(penguin)	bird(swan)
	bird(hawk)		
負例：	bird(bat)	bird(frog)	bird(snake)
背景知識：	fly(owl,true)		fly(penguin,false)
	fly(swan,true)		fly(hawk,true)
	fly(bat,true)		fly(frog,false)
	fly(snake,false)		cover(owl,feather)
	cover(penguin,feather)		cover(swan,feather)
	cover(hawk,feather)		cover(bat,none)
	cover(frog,none)		cover(snake,scale)
	egg(owl,true)		egg(penguin,true)
	egg(swan,true)		egg(hawk,true)
	egg(bat,false)		egg(frog,true)
	egg(snake,true)		

図 1: 目標概念 bird(X) の正・負例と背景知識

3 統計値に基づく MSC 中のリテラルの削除

上で述べた探索手法を用いた場合，探索空間の大きさが MSC のリテラル数 n が大きくなると指数関数的に大きくなる事に注目すると，n を減らす事で，探索空間を縮小し，探索時間を削減することが可能である．

そこで，本稿では，統計的観点から MSC のリテラルの中で仮説に用いられる可能性が少ないと考えられるリテラルを削除する手法を提案する．

3.1 出現率に基づく削除候補の同定

本手法ではまず，背景知識中の属性 a に関してその属性値 vが出現する頻度に着目する．ここで，正例における属性 a の属性値 vの出現率 $P_{pos}(a, v)$，負例における属性 a の属性値 vの出現率を $P_{neg}(a, v)$ は以下のように求められる．

$$P_{pos}(a, v) = \frac{\text{属性 } a \text{ に関して値 } v \text{を持つ正例数}}{\text{全正例数}} \quad (2)$$

$$P_{neg}(a, v) = \frac{\text{属性 } a \text{ に関して値 } v \text{を持つ負例数}}{\text{全負例数}} \quad (3)$$

例えば，図 1の属性 flyとその値 $true$ に関しては，$P_{pos}(fly, true) = 0.75$，$P_{pos}(fly, true) = 0.33$ である．

このような出現率を考えた場合，仮説に用いられる可能性の低いリテラル $L(= a(d, v))$ としては，$P_{pos}(a, v)$ と $P_{neg}(a, v)$ の値が，共に，ある高い閾値 (以下，αと

Reducing Hypothesis Space Based on Statistical Information in Inductive Logic Programming
Yosuke KURODA, Kouzou OHARA, Noboru BABAGUCHI, and Tadahiro KITAHASHI
I. S. I. R., Osaka University

する) 以上となるよう L が挙げられる．何故なら，属性値 vはほとんどの事例が持つ性質であり，正例と負例を区別するための仮説に用いられる可能性が低いと考えられるからである．

3.2 相関関係の強いリテラルの選別

次に属性の出現率に基づいて同定した削除候補リテラルのうち，実際には最適仮説の条件部となり得るものまで削除しない為に，削除候補リテラル L_1 と非削除候補 L_2 の組合せに関して相関関係を調べる．具体的には，L_1 と L_2 各々における事例名を変数化したリテラル L_1' と L_2' に関して，L_2' を条件部とする仮説の評価値より，L_2' と L_1' の連言を条件部とする仮説の評価値の方が高ければ，L_1' が表す属性値は L_2' を満たす正例と相関関係が強く，逆に負例とは相関関係が弱いと判断する．従って，L_2' を条件部にもつ仮説に L_1' を加える事で，より高い評価値をもつ仮説が得られる事が期待できるので，L_1' の元となる L_1 を削除候補から除外する．

例えば，評価関数として下記の式 (4) が与えられ，式 (1) の MSC においてリテラル $egg(owl, true)$ を削除候補とした時に，$fly(X, true)$ のみを条件部とした時の仮説の評価値 $f_1 = 0.81$ に対して，$egg(X, true)$ と組合わせた時の評価値は $f_2 = 1 > f_1$ となるのでリテラル $egg(owl, true)$ は削除候補から除外される．

$$f(p,n,c) = (p-n-c) \times (1 - H(\frac{p}{p+n})) \quad (4)$$
$$(H(x) = -x\log_2(x) - (1-x)\log_2(1-x))$$

p:仮説により被覆される正例数
n:仮説により被覆される負例数
c:仮説の条件部の述語数

3.3 統計値に基づくリテラルの削除のアルゴリズム

以上の議論から，統計値に基づいた MSC 内のリテラルの削除を形式的にまとめると以下のようになる．ここで Bは背景知識，Fは仮説 Hの評価関数，d は事例名，L，L'はリテラル，a，vは L の属性，属性値，H_L，$H_{LL'}$はそれぞれ条件部が L，L と L'の連言から生成される仮説である．

1 $S_1 = \{L | L = a(d,v) \in B,$
$P_{pos}(a,v) \geq \alpha, P_{neg}(a,v) \geq \alpha\}$
2 $S_2 = \{L | L \in S_1, L' \in (B - S_1), F(H_{L'}) \geq F(H_{LL'})\}$
3 MSC から S_2に含まれるリテラルを削除する．

4 評価実験

提案手法を我々の研究グループが提案した ILP システム G-REX[3] 上に実装し，評価実験を行った．評価関数には式 (4) を用い，実験対象としては，UCI で公開されているマッシュルーム DB を用いた．マッシュルーム DB とは，キノコ 8,124 種について，毒の有無を 22 の属性で記述したデータベースである．各属性は 2 個から 12 個の属性値をとる．よって背景知識は 8,124 × 22(約 17 万) 個のリテラルからなる．本実験では目標概念を「毒を持つ」($poison(A)$) とし，その場合の正例数は 3,916 個，負例数は 4,208 個となる．実験内容としては，提案手法の有効性を示す為の比較に，MSC のリテラルを削除せずに学習する実験 1 と，$\alpha = 0.9$ とした提案手法を実装して学習する実験 2 を行った．

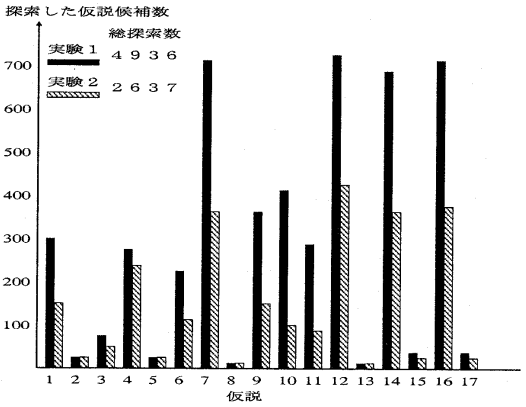

図 2: 生成した仮説数と探索した仮説候補数

実験結果を図 2に示す．生成された仮説は 17 個であり，実験 1，2 のいずれにおいても同じ仮説が生成された．図 2から本手法を用いることで仮説探索の効率が大幅に改善されていることが分かる．具体的には，実験 1 の総探索数が 4,936 個に対して，実験 2 では総探索数は 2,637 個に減っており，実質，約 47%の探索数削減に成功している．

5 まとめ

本稿では，統計値に基づいた ILP における仮説探索の効率化手法を提案した．本手法では MSC のリテラルの削除による仮説空間の縮小を，出現率及びリテラル間の相関関係から，仮説の述語候補になる可能性の低いリテラルを削除することで実現し，実験によってその有効性を示した．提案手法は，大規模なデータベースからの仮説獲得時など，探索数が膨大になる場合において有効であると考えられる．

今後の課題としては，本手法の統計的評価，及び対象とする背景知識の形式の多様化などが挙げられる．

参考文献

[1] 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,pp.655-664(1997).

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

[3] 高，大原，馬場口，北橋：例外関係に着目した不完全知識の獲得システムの実装と実験的評価, 人工知能学会研究報告 (SIG-FAI-9803-3),pp.11-17(1998).