

帰納論理プログラミングを用いた例外を含む知識の学習の並列化

6P-7

大西 悠季生 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

帰納論理プログラミング (Inductive Logic Programming: ILP)[1] を用いた知識の学習は、目標概念の正例、負例、および背景知識が与えられたときに、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、仮説候補からなる探索空間から見つけ出すものである。この ILP における知識の学習を、例外を含む一般的な知識をも学習できるように拡張したシステムの研究が数々の研究者によって進められている [2, 3]。例外を含む知識は可読性、及び一般性が高い反面、未知の例外に対する予測精度が低いという側面をもつ。従って、学習された知識の例外に関する知識を再帰的に学習する必要があり、その学習時間が学習時間全体に対して大きな割合を占めるといふ、従来の ILP には無かった問題が生じる。そこで本稿では、例外を含む知識の学習を並列化することで学習時間の短縮を図る。

2 例外を含む知識の学習

図 1 に示すような正例、負例、背景知識が与えられた場合、例外を含む知識の学習では、まず初めに、

$$fly(X) \leftarrow bird(X) \quad (1)$$

のように、より多くの正例を導き、可能な限り負例を導かない仮説候補を最適仮説として獲得する。このステップを、ここでは一般学習と呼ぶ。次に予測精度を上げるために、獲得された最適仮説の被覆する負例 $fly(emu)$ を例外として、例外に関する知識の学習を再帰的に実行する。その結果、

$$\neg fly(X) \leftarrow bird(X), penguin(X) \quad (2)$$

を獲得するが (ただし \neg は否定記号)、この仮説にも例外 $fly(ping)$ が含まれるため、同様に再帰処理が適用され、仮説

$$fly(X) \leftarrow bird(X), penguin(X), super-penguin(X) \quad (3)$$

が獲得される。ここまでの例外に関する知識の学習を、ここでは詳細学習と呼ぶ。詳細学習が終了すると、再び最適仮説を求める一般学習が実行される。従って、例外を含む知識の学習は、図 2(a) に示すように一般学習と詳細学習とが交互に繰り返されているものと捉えることができる。

3 システムの並列化

3.1 2つの学習問題の独立性

ここで、先の例において、一般学習で生成された仮説 (1) に基づく詳細学習は、仮説 (1) のクラスに対し

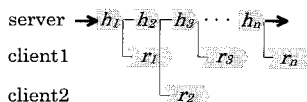
正例: $fly(haky) \quad fly(ealy) \quad fly(tweety)$
 $fly(ping)$
 負例: $fly(emu) \quad fly(leo)$
 背景知識: $bird(haky) \quad bird(ealy)$
 $bird(tweety) \quad penguin(emu)$
 $super-penguin(ping) \quad animal(leo)$
 $animal(X) \leftarrow mammal(X)$
 $animal(X) \leftarrow bird(X)$
 $bird(X) \leftarrow penguin(X)$
 $penguin(X) \leftarrow super-penguin(X)$

図 1: 目標概念 $fly(X)$ の正・負例と背景知識

てその例外のクラス (仮説 (2))、さらにその例外のクラス (仮説 (3))、というクラス階層の生成に相当していることに着目する。一般に、このようなクラス階層における任意の例外クラスは、その上位クラスの部分クラスであることから、クラス階層の生成は、最上位クラスが確定した時点で、当該クラス中の事例のみを対象としてより特殊な概念記述を学習するという個別の学習問題と捉えることが可能である。従って、一般学習によってある最適仮説 H が求まると、 H に基づく詳細学習は独立して行うことができる。

$$\rightarrow h_1 - r_1 - h_2 - r_2 - h_3 - r_3 \dots h_n - r_n \rightarrow$$

(a) 従来の処理



(b) 並列化による処理

h_n 一般学習, r_i 詳細学習

図 2: 学習過程の概念図

3.2 server-multi-client 型システムによる例外を含む知識の学習

前節での議論から、一般学習によって最適仮説 H が求まると、 H に基づく詳細学習は、他の一般学習、詳細学習とは独立して行える。すなわち、各詳細学習を図 2(b) に示すように別々のプロセッサに依頼し、並列化させることで学習時間の短縮が可能となる。

このような並列化を実現するために、提案システムでは、ネットワーク上の複数のプロセッサが LAN などによって接続されている環境を仮定し、それらのプロセッサのうち 1つを一般学習を担うサーバ、残りのプロセッサを詳細学習を担うクライアントとして利用する。また、サーバは各クライアントの 3 状態 { 準備状態, 空き状態, 実行状態 } を管理するクライアントテーブルを持ち、タスクを依頼するクライアントの選択、及び学習の終了判定に用いる。

Parallel Processing of Hypothesis Search
in Learning Default Rules
Yukio ONISHI, Kouzou OHARA, Noboru BABAGUCHI
and Tadahiro KITAHASHI, I.S.I.R., Osaka University

一方、提案システムにおけるサーバ、クライアント間でやり取りされる主なメッセージは、図3に示すようにサーバからクライアントへの詳細学習の依頼と、クライアントからサーバへの終了報告であり、それぞれのメッセージに付随してサーバが獲得した最適仮説、及びクライアントの学習結果を転送する。なお、サーバに入力される正例、負例、及び背景知識に関しては、学習開始時に各クライアントに転送するものとする。

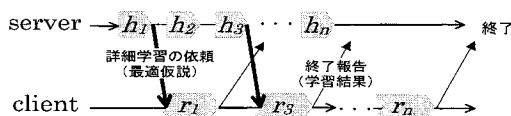


図3: サーバ/クライアント間のメッセージ

3.3 多重並列化

クライアントにおける詳細学習も、図4に示すように、与えられた事例集合における一般学習 $\{rh_1, rh_2, \dots, rh_n\}$ と、詳細学習 $\{rr_1, rr_2, \dots, rr_n\}$ の2つの独立した学習問題に分割できるため、各クライアントの仮説探索をさらに並列化することが可能となる。すなわち詳細学習を多重に並列化させて、空き状態のクライアントを効果的に利用することができる。具体的には、あるクライアントで生成された仮説 H が例外を被覆する場合、それを説明するための詳細学習を行うが、これを他の空いているクライアントに依頼することで、多重並列化を実現する。

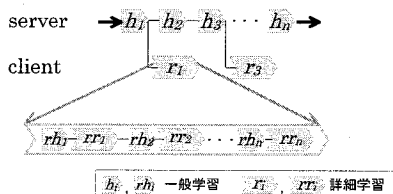


図4: 詳細学習の中の一般学習と詳細学習

4 評価実験

提案手法に基づき筆者の所属する研究室がこれまで開発してきた例外を含む知識の学習システムであるG-REX[3]を並列化し、提案手法の有効性を実験的に検証した。実験では、UCIで公開されている機械学習用のデータベースであるマッシュルームデータベースを対象に、全データ、及び正例・負例をそれぞれ1,000個ずつ無作為に抽出したデータ10組に対して、それぞれ並列化しない場合 (alone), クライアント2台で並列化した場合 (2c), クライアント2台で多重並列化した場合 (2c-m) について学習実験を行った。使用したデータベースの事例数と背景知識の数を表1に、用いたプロセッサの仕様を表2に示す。なお、並列化しない場合の学習は、サーバ上で行った。

結果を表3に示す。全データで実験した結果を“origin”, 10組のデータベースを用いた実験結果の平

表1: 使用したデータベースの事例数と背景知識の数

目標概念	正例数	負例数	背景知識の数
毒をもつ	3,916	4,208	約17万

表2: 用いたプロセッサ

用途	プロセッサ	CPU	Memory
サーバ	SUN SS20	hyperSPARC 125MHz	352MB
クライアント	SUN SS5	microSPARCI 85MHz	128MB

均を“AVG”, 10組のうち並列化による効果が最大のものと最小のものをそれぞれ“MAX”, “MIN”と表記する。

表3: 実験結果

DB		alone	2c	2c-m
origin	学習時間 (sec)	3,615	2,437	2,348
	速度向上比	1.00	1.48	1.54
AVG	学習時間 (sec)	585.9	411.6	410.6
	速度向上比	1.00	1.43	1.43
MAX	学習時間 (sec)	654	396	399
	速度向上比	1.00	1.65	1.64
MIN	学習時間 (sec)	517	455	457
	速度向上比	1.00	1.14	1.13

表3から分かるように、提案手法では、全マッシュルームデータベースに対して約1.5倍の効率化が確認できたが、多重並列化による効果はわずかであった。これは、使用したデータから生成されるクラス階層の深さが、高々2であったため、多重に並列化される回数がほとんどなかったことが原因と推測される。一方、10組のデータの実験における速度向上比の平均は約1.4倍であったが、データセットによって1.1倍~1.7倍とばらつきが見られた。MINのように効果がほとんど見られなかった原因としては、一般学習に要する時間が大きく、クライアントが効率よく利用されなかったことが考えられる。

5 まとめ

本稿では、例外を含む知識の学習における仮説探索の並列化手法を提案した。提案手法では、例外に関する知識の学習を一般学習と、例外に関する学習である詳細学習に分割し、詳細学習を他のプロセッサで処理することで並列化を実現し、実験によりその有効性を確認した。本手法は、例外の多い複雑かつ多用な実世界のデータに対して有効であると考えられる。

今後の課題として、最適仮説の生成を並列化する手法との組み合わせや、各プロセッサの能力に応じた詳細学習の割り振りによる更なる効率化が挙げられる。

参考文献

- [1] S.Muggleton: Inductive logic programming, New Generation Computing, Vol.8, pp.295-318(1991)
- [2] E. Lamma, F. Riguzzi: Pereira Strategies in Combined Learning via Logic Programs, Machine Learning, Vol. 38, Number 1/2, pp.63-87(2000)
- [3] 高秀幸, 大原剛三, 馬場口登, 北橋忠宏: 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)