

KL1 処理系による並列帰納学習*

5R-1

溝口 文雄 川野辺 甲 大和田 勇人†

東京理科大学 理工学部‡

1 はじめに

近年、機械学習の分野で帰納的推論を取り入れ、背景知識と事例から一般化規則を導出する際に、用いられた概念を論理プログラムで生成する研究が盛んに行なわれている。その中でも、MuggletonのGOLEM[1]などが代表的である。この帰納学習システムは主に、エルブラン空間における離散的な値を持つ事例を学習する。これに対して、エルブラン空間のみならず、ユークリッド空間における実数領域に含まれている事例をも学習することが可能な、関数従属性を用いた帰納学習システム [2] がある。

この帰納学習システムは、関数従属性の考えに基づいて、与えられた事例よりいえる可能な限りの全ての規則を生成するので、事例数の増加に伴って実行時間も増加するため、あまり効率が良いとはいえない。

そこで本稿では、実行時間を短縮し効率を上げるために、関数従属性を用いた帰納学習システムに並列処理を導入し拡張した並列帰納学習システム PILS(Parallel Inductive Learning System) について述べる。

このPILSを構築するために採用した、KL1 処理系上で稼働する並列論理型言語 KL1 は、逐次的な考えから出発した言語ではなく最初から並列処理を意識した言語で、データフロー同期機構やプラグマによる物理的な並列実行の指定などを用いることによって、非常に簡単に並列処理を行なうことができる。そのため、並列機能学習システムを構築する上で非常に適した言語であるといえる。

2 並列帰納学習システム (PILS)

2.1 並列性の導入

関数従属性を用いた帰納学習システムは、ユークリッド空間における実数領域に含まれる事例を学習する際に、

可能な限りの全ての事例に対する組合せを考慮する。そして、考えられるだけの規則を導出することによって、より有効な一般化規則を導出する。(図1)

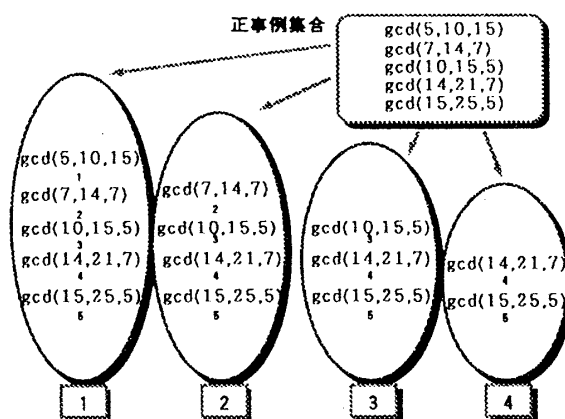


図1: 正事例集合からの事例の抽出

具体的には図1に示されているように、正事例集合の事例数が5つならば、与えられた正事例集合からその入力順に事例を抽出し4つのグループを作る。そして、そのグループの上から一番目と二番目の事例に対しRLGG, 関数従属性によるリテラルの抽出などを行なって仮説を生成する。しかし、このような方法を逐次的に処理したのでは、事例数の増加に伴って、計算時間が増加し、効率の良い仮説の生成が困難になる。そこで、本システム PILS は、図1の1から4のグループをプロセッサに割り付け、並列処理を行なうことで計算時間を短縮し、効率の良い仮説の生成を可能にしている。

並列処理による高速化が期待できるのは、図1の1から4の各グループがそれぞれ独立であり、各グループは他のグループに関して無関係に探索を行なえるからである。

2.2 最良一般化

PILS では上で述べたような負化分散を行ない、与えられた事例よりいえる全ての仮説を各プロセッサ上で求めた後、その仮説をあるプロセッサでマージする。そし

*Parallel inductive learning by KL1 language processor

†Fumio Mizoguchi, Masaru Kawanobe, Hayato Ohwada

‡Faculty of Sci. and Tech., Science Univ. of Tokyo

て、その仮説の中からより一般性のある規則を抽出するために、最良一般化を行なう。これは、「一つの事例が二つ以上の規則に含まれている場合、他の事例をより多く含んでいる規則をその事例を含む最良の一般化規則とする」というものである。

2.3 PILS システムの流れ

PILS システムの流れは以下ようになる。ここで Step3 から Step4 までは並列に実行される。

[Step1] 正事例の集合を ε^+ 、背景知識を K とする。ここで、正事例集合（背景知識集合）に含まれている事例には、入力された順に e_1, e_2, \dots, e_m というように 1 から m まで番号が付けられる。

[Step2] $e_{i,j} \in \varepsilon^+$, $e_{i,j+1} \in \varepsilon^+$ である $e_{i,j}, e_{i,j+1}, (i = j, i = 1, \dots, n-1, j = 1, \dots, m-1)$ を選択し、それぞれのボディ部を $e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1} \in K$, $e_{i,j}, e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1} \in K$ ($k = 2, \dots, j$) とする。ここで、 $e_{1,1} \in \varepsilon^+$ のボディ部は \square とする。そして、以下の Step3 から Step4 の繰り返しを $i = 1$ から $m-1$ まで並列に実行する。（ここで i は図1の横方向のグループ番号を表しており、 j は図1の縦方向の事例の番号を表している。

[Step3] $C_1 = e_{i,j} \leftarrow e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1}$, $C_2 = e_{i,j+1} \leftarrow e_{i,j}, e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1}$ として $rlgg(C_1, C_2)$ を求め、関数従属性によるリテラルの抽出により節を決定し、それを仮説 C とする。

[Step4] もし、関数従属性によって節が決定しないのであれば、仮説を $e_{i,j} \leftarrow e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1}$ と $e_{i,j+1} \leftarrow e_{i,j}, e_{i,j-1}, e_{i,j-2}, \dots, e_{i,j-k+1}$ に分け、それぞれ次の事例 $e_{i,j+2} \leftarrow e_{i,j+1}, \dots, e_{i,j-k+1}$ と Step3 の処理を行なう。

そうでなければ、得られた仮説 C と次の事例 $e_{i,j+2} \leftarrow e_{i,j+1}, \dots, e_{i,j-k+1}$ に関して Step3 の処理を行なう。

[Step5] $j = 1, \dots, m$ まで Step3 から Step4 まで繰り返す。繰り返しが完了したら最良一般化を行ない、得られた仮説の中でより一般性のある規則を残していき、最終的にそれらを一般化規則とする。

3 評価

図2は、ICOTの並列推論マシン Multi-PSI(16PE版)上で、ユークリッド空間における学習例として factorial, fibonacci, gcd を実行させた時のそれぞれの速度比の平均とプロセッサ数の関係を表したものである。速度比は、プロセッサ数が一つの時の実行時間を1として、プロセッサ数を増加させた時に何倍になったかを表したものである。

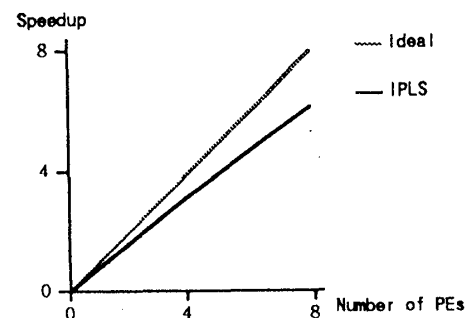


図2: PILSの台数効果

図2から、プロセッサ数の増加に伴って良好な台数効果が得られたので、PILSの学習アルゴリズムに含まれる並列性が高いことがわかる。従って、事例数が増加した場合には、並列推論マシン(Multi-PSI, PIM)のプロセッサ数の増加による仮説探索の効率化が期待できる。

4 まとめ

関数従属性を用いた帰納学習システムでは、事例数の増加に伴って実行時間が増大し、実行効率が悪くなるので、本稿では並列処理の考えを導入して、実行時間の増加を防ぎ、実行効率を上げることができる並列帰納学習システム(PILS)について述べた。

また、PILSのアルゴリズムは、負荷分散するための部分問題が各部分問題に対して独立であり、無関係に実行されることが可能であるため、非常に高い並列効果が期待できるのである。

参考文献

- [1] S.Muggleton, C.Feng: Efficient induction of logic programs. *In New Generation Computing*, 1991, Vol8, pages 42-61, 1990
- [2] 大和田, 溝口: 制約論理プログラムの帰納汎化システムの設計, 情報処理学会第48回全国大会, 1994