

並列分散型帰納学習システムの設計*

7P-5

川野辺 甲 大和田 勇人 溝口 文雄†
 東京理科大学 理工学部‡

1 はじめに

近年、機械学習と論理プログラミングの融合である帰納論理プログラミングの研究が盛んに行われており、種々のシステムが開発されている。しかし、このようなシステムは、大規模な有限メッシュ分割法を求めるような事例の場合には、妥当な時間で学習することができないといった問題がある。そこで、従来の帰納学習の枠組に並列処理技術を導入した並列帰納学習システムが作成された。[1]しかし、このシステムでは種類の違う事例を同時に数多く学習するには都合が悪い。そこで、本稿では帰納学習自体を分散して一般化規則を求める機構を備えた並列分散型帰納学習システムを設計し、その有効性について示す。また、並列帰納学習システムは専用の並列マシン上で動作したのに対し、並列分散型帰納学習システムは、汎用性のあるPVM(Parallel Virtual Machine)を用いた分散版KLIC処理系上で動作する。

2 並列分散型帰納学習

2.1 帰納学習の並列性

一般に帰納学習の研究は、帰納論理プログラミングの枠組で考えられている。帰納論理プログラムは、正事例集合を \mathcal{E}^+ 、負事例集合を \mathcal{E}^- 、背景知識集合を \mathcal{K} としたときに、 $\mathcal{K} \cup \mathcal{H} \vdash \mathcal{E}^+$ 及び $\mathcal{K} \cup \mathcal{H} \not\vdash \mathcal{E}^-$ を満足する一般化規則 \mathcal{H} を求めるものである。

この時、仮説は論理プログラムの形で、帰納的一般化によって求められる。一般には、与えられた正事例集合 \mathcal{E}^+ の要素の全てにおいて一般化規則を求めるのではなく、 \mathcal{E}^+ の部分集合の下で一般化規則を求めることになる。従って、正事例集合の部分集合ごとに独立して帰納的一般化を行ってもよいということになる。

2.2 並列分散型帰納学習の枠組

並列分散型帰納学習とは、前出の帰納学習の内在的並列性に着目して、その部分を並列に実行するといった方法と、帰納学習自体を分散させて実行するといった方法を合わせた帰納学習方式である。前者を領域分割による並列分

散型帰納学習、後者を機能分割による並列分散型帰納学習という。そこで、並列分散型帰納学習は以下のような流れで行われる。

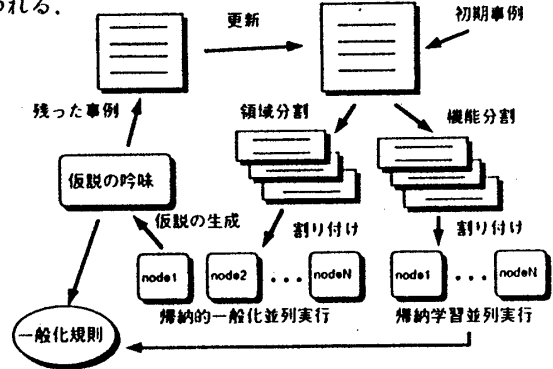


図1: 並列分散型帰納学習の流れ

まず、与えられた初期事例を分割し、機能分割による場合は、各ノードに分散して帰納学習を行い、得られた仮説を集めて一般化規則集合を得る。領域分割による場合は、各ノードに分散して1個ずつ仮説を生成し、生成された仮説を吟味して冗長な仮説を削減する。そして、残された仮説を仮説集合に入れる。その時、その仮説に含まれない事例が残らなければこの帰納学習のサイクルは終了し、事例が残った場合には、さらに残った事例を新たな事例集合として、帰納学習のサイクルを繰り返す。結局、最終的に得られた仮説集合が一般化規則集合となる。

3 並列分散型帰納学習システムの設計

以上のような並列分散型帰納学習の枠組に基づいて並列分散型帰納学習システムを設計する。このシステムの構成は図2のようになる。ホストマシンの入出力部から得られた事例を負荷分散部で分割、負荷分散し、各マシン上の帰納学習部で帰納学習される。そして、学習された結果をマージし、冗長な節を取り除き最終的な一般化規則を得る。PVMは、UNIXマシンをネットワーク結合させ、その集合を疑似並列マシンのごとく使用するソフトウェアである。KLICは並列論理型言語 KL1 を UNIX 上で実行するための処理系であり、並列論理型言語 KL1 は、負荷分散指定や同期処理が従来の並列言語よりも簡単にできる並列ソフトウェア研究のための最適なツールである。KLICとKL1は、第五世代コンピュータプロジェクトで開発されたものである。UNIXは、SUN SPARC station IPXを用いた。

*Design of a Parallel and Distributed Inductive Learning System

†Masaru Kawanobe, Hayato Ohwada, Fumio Mizoguchi

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

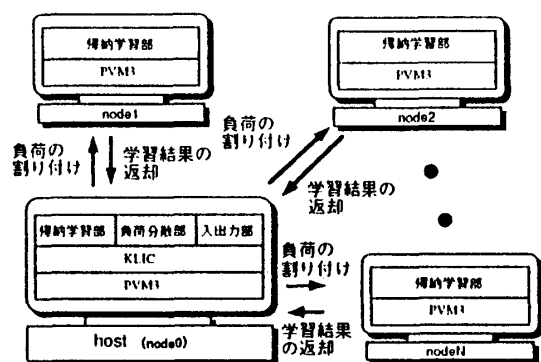


図 2: 本システムの構成

4 例題

ここでは、例題として有限要素メッシュ分割問題へ本システムを適応した結果について述べ、本システムの有効性を示す。有限要素メッシュ分割は、ある物体にかかる圧力を分析するために機械工学のエンジニアなどが良く用いる方法である。この時、どのようにメッシュ分割を行えば良いかという問題が生じる。そこで、本稿では分割を行うための規則を並列分散型帰納学習システムを用いることで導出する。背景知識には Dolsak の研究 [2] によるデータを用いた。表 1 に正事例数 76 個、背景知識 618 個、負事例 60 個の事例を用いて領域分割による並列実行を行った結果と正事例を a, b, c の三つの構造別に分け、それぞれの構造別に分割し実行した結果を示す。

表 1: 例題の結果

領域分割による並列分散型帰納学習の実行結果				
ノード数	1 node	3 nodes	5 nodes	10 nodes
実行時間 (msec)	147842	59501	65648	131020
台数効果	1.00	2.48	2.25	1.14
機能分割による並列分散型帰納学習の実行結果				
実行時間 (msec)	102915	71121		
台数効果	1.00	1.45		

領域分割による結果をみると、ノード数 5 の付近で徐々に並列実行効率が低下している。これは、図 1 で示したように、本システムでは仮説を作る 1 サイクルの内、逐次実行では規則を 1 個、ノード数 n の並列実行では規則を n 個導出するからである。つまり、並列実行は、帰納的一般化を行う実行時間は減少するが、仮説を吟味するための実行時間が増加するため、結果的に減少した帰納的一般化の実行時間よりも増加した仮説吟味のための実行時間の方が大きくなってしまったからである。また、PVM はノード間通信にかかる時間が ICOT の並列推論マシン PIM など比べて 2 桁遅いので、ノード間通信によるオーバーヘッドを補うだけの分散効果がなかったからとも考えられる。一般に、使用ノード数の最適値は、対象問題の構造や性質に依存す

る。従って、この問題では、効果的な並列実行を行うためにはノード数 4~5 が有効であるといえる。機能分割による結果を見ると、機能分割による並列実行は、領域分割による並列実行よりも台数効果が低い。これは、構造別に分散した事例の数と同じではないので、負荷が均等に分散されなかったことが原因である。さらに、PVM を用いた分散版 KLIC 処理系では、一つの大きなプログラムを他ノードに分割して実行する際、プログラムコピーのオーバーヘッドが起り、プログラムの一部を分割して実行するよりかなり多くの時間がかかってしまうことが原因であるとも考えられる。

また、領域分割によって得られた規則

$$\text{mesh}(A, 8) : -\text{circuit}(A)$$

は、機能分割では、構造 c でのみ求めることができた。このことから、この規則は他の構造には適応できず、構造 c のみに適応できる規則であることが分った。このように、事例を構造別に分割して規則を求める事は意義のあることであり、その場合に本システムを適応することは有効であるといえる。

5 まとめ

本稿では、帰納学習に並列処理の技術を融合させた並列分散型帰納学習という帰納学習の新たな枠組を提案した。そして、PVM を用いた分散版 KLIC 処理系上で動作する並列分散型帰納学習システムを設計し、その有効性を示した。有限メッシュ分割問題では、ノード数 4~5 の並列実行が効果的であり、構造別に一般化規則を求めることは意義のあることで、本システムを適応することが有効であることが分った。また、このような帰納学習問題に対して、最適なノード数を並列実行前に解析する手法を求めることは非常に重要であり、今後の課題である。

謝辞

本研究は、第五世代コンピュータプロジェクト (ICOT) の共同研究プロジェクト KL1 による制約処理の研究の一環として行われた。本研究を進める上で、分散版 KLIC 処理系に関して貴重な助言をして下さった近山隆氏ら ICOT の研究者の皆様へ深く感謝致します。

参考文献

- [1] 溝口, 川野辺, 大和田: KL1 処理系による並列帰納学習, 情報処理学会第 48 回全国大会, 1994
- [2] B. Dolsak, S. Muggleton: The Application of Inductive Logic Programming to Finite-Element Mesh Design. In *New Generation Computing*, 1991, Vol8, pages 453-472, 1990.