TR-689

推定木学習アルゴリズムの並列化方式

中莖　洋一郎、古関　義幸、山中　みどり

September, 1991

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5
Telex ICOT J32964

Institute for New Generation Computer Technology

推定木学習アルゴリズムの並列化方式

中室洋一郎　古関義幸　田中みどり

日本電気(株) C&Cシステム研究所

1　はじめに

近年、多数の事例を基に帰納的に学習を行なう方式の研究が盛んに行なわれている。しかしながら、一般にそれらの計算量は非常に大きなものとなる傾向があるため、計算量を減らすために様々なヒューリスティックの導入等が行なわれている。これらの問題は、基本的に探索問題と捉えることができるため、並列処理を行うことによる高速化が期待される。探索空間の分割・並行処理による効果の他に、並列処理の効果として、問題によっては、ある部分の探索によって得られた情報が、他の部分の探索の効率化に寄与することが期待される。本報告では、推定木の学習[1, 2]を対象として、局所的、及び大域的な情報に基づく分枝限定法による、分散・協調的な並列探索アルゴリズムを提案する。

2　推定木の学習問題

ここで、問題の定式化を行う。まず、排他的かつ網羅的な事象の集合 $X = \{x_1, x_2, \cdots x_m\}$ を考える。さらに $A = \{a_1, a_2, \cdots a_n\}$ を考え、各属性 a_j $(j = 1, 2, \cdots n)$ の取り得る値を有限集合 $Dom(a_j)$ とする。表1に示すように各 x_i には、それぞれ属性 a_j $(j = 1, 2, \cdots n)$ に対する属性値 v_{ij}、各事象 x_i が過去の試行で生起した頻度 n_i が与えられる。

表1　与えられる観測結果

事象	属性				頻度
	属性 a_1	属性 a_2	⋯	属性 a_n	(回)
x_1	v_{11}	v_{12}	⋯	v_{1n}	n_1
x_2	v_{21}	v_{22}	⋯	v_{2n}	n_2
x_3	v_{31}	v_{32}	⋯	v_{3n}	n_3
⋮	⋮	⋮	⋱	⋮	⋮
x_m	v_{m1}	v_{m2}	⋯	v_{mn}	n_m

このような観測データを基に、将来各事象の起こる確率を推定する方法を推定木という形で表現することができる[1, 2]。推定木の例を図1に示す。

各分岐点●にはそれぞれある属性 a_i が対応し、(図では属性 a_1, a_2)、その子への枝にはそれぞれ属性 a_i の取り得る値の部分集合 $A_{i1}, A_{i2}, \ldots A_{il}$ (l はその分岐点からその子への枝の本数)が対応する。

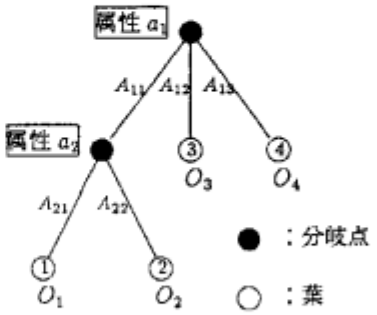

図1　推定木

与えられた事例に対して、考えられる推定木の数は膨大である。その中から「将来起こる事象の予測に最適な」推定木を選択するための基準としてMDL基準を用いる。推定木の記述長は、次に示す $L1 + L2$ で計算され、この値が最も小さい木を探すことが問題である。

$$L1 = \sum_{x \in P \cup Q} \log(n - d_x + 1) + \sum_{x \in Q} \frac{1}{2} \log O_x + \sum_{x \in P} \{\log(k_x - 1) + \log c1(k_x, l_x)\}$$

ここで、Pは全ての分岐点の集合、Qは全ての葉の集合であり、分岐点 x に対して l_x は分岐数、d_x は根からの深さ(根を0とする)を示す。さらに $k_x = |Dom(a_i)|$ (ただし、a_i は分岐点 x に対応する属性)である。また、$c1(k_x, l_x)$ は、$l_x < k_x$ の時には ${}_{k_x}C_{l_x-1}$、それ以外の場合は1である。

一方、L2は、次の式で計算される。

$$L2 = -\sum_i n_i \log \hat{p}_i$$

ただし、$\hat{p}_i$ は、そのモデルによって推定される事象 x_i の生起確率である。

3　並列探索アルゴリズム

まず、効率的な探索を行うため、L1とL2の性質について考察する。L1は、直観的には木の複雑さを示す指標であり、木が大きくなる程大きな値をとる。より正確に分析するため、ある木T1の1個以上の葉を部分木に置き換えて得られる木T2に対して、T2≻T1という順序関係を定義する。このときにL1の定義より、明かにL1(T1)≤L1(T2)が成り立つ。また、L2に関し

A Parallel Algorithm for Learning Presumption tree
Y. Nakakuki, Y. Koseki and M. Tanaka
C&C Systems Research Labs., NEC Corporation

ては逆に L2(T1)≥L2(T2) が成り立つ。さらに、半順序関係 $\succ$ に関して最大の任意の木 T において L2 は最小値 L2(T)=$L2_{MIN}$ をとることが知られている。これらの性質を用いて、考えられる全ての推定木 T の中で最も L1(T)+L2(T) が小さくなるものを効率よく探す。

3.1 局所的な情報を用いた枝刈り

探索の順序としては、順序関係 $\succ$ に関して小さい推定木から先に行う。従って、最初は、分岐のない 1 ノードのみからなる推定木の長さが計算される。また、ある推定木 T の長さ調べる前に、それより小さいすべての推定木について調べられることになる。探索の途中状態では、次に探索可能な推定木は多数存在するのが普通である。逐次型の処理では、それらを順次調べることになるが、ここでは並列マシン Multi-PSI (16PE) を用いて並列に探索する方式を採用する。

さらに効率化を図るため、ここで、ある木 T の長さ L1(T)+L2(T) を計算した後に、T$\succ$T' なる木 T' の長さを調べる必要があるかどうかを考える。L1 の定義より、

$$L1(T')-L1(T) \geq \log\,(n-d_x+1)+\log(k_x-1)+\log\ c1(k_x,l_x)$$

であることが判る。一方、L2 に関しては、

$$L2(T') - L2(T) \geq L2_{MIN} - L2(T)$$

であるから、2 つの不等式の右辺の和が正であるとすると、T' の長さは T の長さよりも大きくなることが判る。従って、そのような T' については、調べる必要がなくなる。そこで、このような性質に基づいた分枝限定法を採用する。

3.2 大域的な情報を用いた枝刈り

前章では、ある推定木を探索した時に、それより大きな木の探索を行う必要があるかどうか、判断する方法について述べた。この処理は、探索を並列に行う場合に、各々が独立に判断していくものであった。これに対して、処理の途中で、ある推定木 T_0 の長さがかなり短いという情報が得られたとすると、他の探索において、T_0 よりも明らかに記述長の長い木の探索を行う必要がなくなる。以下に、その具体的な処理方法を示す。

上記の木 T_0 に対して、前章の $T'(\succ T)$ に関して、

$$L1(T') + L2(T') \geq L1(T') + L2_{MIN} > L1(T) + L2_{MIN}$$

が成り立つ。ここで、もし

$$L1(T) + L2_{MIN} \geq L1(T_0) + L2(T_0)$$

であるとすると、上の 2 つの不等式より、T' は T_0 より短くならないことがわかるため、T より大きな推定木については、もはや調べる必要がなくなることがわかる。

4 実験結果

実験は、疎結合マシン Multi-PSI (16PE) 上で行った。実験に用いたデータは、属性数 5、事例数 100 程度の入力を対象としている。実験プログラムでは、前記の局所的な情報を用いた枝刈りを行っている。実験結果を図 2 に示す。ほぼプロセッサ数に比例して処理効率が上がっており、16 台で約 9 倍程度の高速化が実現されている。現バージョンでは、プロセッサ間の負荷の均等化について、あまり考慮していないため、負荷に関しては多少のばらつきがある。この部分の工夫と、大域的な情報を用いた枝刈り機能によって、さらに高速化が期待される。それらの評価は、今後の課題である。

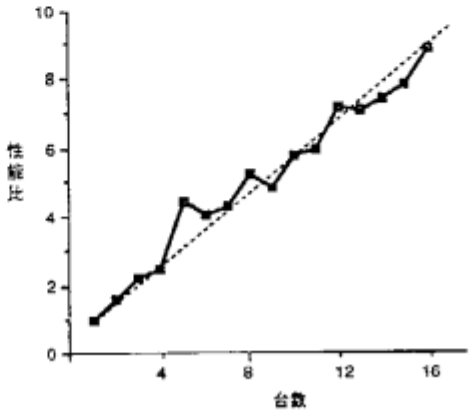

図 2 実験結果

5 おわりに

推定木の学習アルゴリズムの並列化の一手法を提案し、実験結果を示した。このような帰納学習問題においては、効率的な探索が要求されるため、並列処理が有効である。大域的な情報を用いる際には、プロセッサ間の通信量を考慮したインプリメントが要求される。今後、この問題についても研究を進めていく予定である。

謝辞

本研究は、第 5 世代コンピュータプロジェクトの一環として行われたものである。日頃御世話になっている (財) 新世代コンピュータ技術開発機構 新田室長に感謝いたします。

参考文献

[1] Nakakuki, Y., Koseki, Y., and Tanaka, M., "Inductive learning in probabilistic domain," *Proc. AAAI-90*, Vol. 2, pp. 809–814, 1990.

[2] 中亜祥一郎、古関義幸、田中みどり 「確率モデルの学習方式と診断への応用」 情報処理学会研究報告 *(91-AI-74)* Vol. 91 (3), pp. 19–28, 1991.