

2P-6

逐次型帰納学習システムの開発とその応用
分散計算機環境管理支援システムの作成*

大熊 喜之! 大和田 勇人! 溝口 文雄†

東京理科大学 理工学部¹

1 はじめに

現在、ダウンサイジングが叫ばれている。これはハードウェアの進歩によりハード単体の価格性能比が上がったこと、ネットワークが整備されてきたことなどにより、大きなシステムを多くの人で共有するよりも一人で一つの計算機を持ちそれらをネットワークで繋いだ方が全体で見た時に同じ性能を出すのに少ないコストで行なえるようになったからである。

このような流れから、ネットワークに離れた計算機資源が急激に増えてきた。しかし、各計算機の利用法はある程度個人に任されているため、特定の計算機資源にその利用が偏りがちになってしまう。また、これらの管理も専門家ではなく、知識のある人が、本職の片手間でやっているのが現状である。

そのため、管理の専門家でなくても、簡単に管理が行なえるツールが必要である。しかし、従来からあるツール [1] では分散環境全体を管理することはできない。このようなツールの作成を困難にしているのは、調べるべき要素が沢山あり、それぞれが複雑な関係を持っているためである。

そこで、本研究ではそのような要素間の関係を分析するツールとして逐次型帰納学習システムを使用した分析ツールを作成した。ここで利用した学習システムは、逐次的に与えられた数値及び記号データを学習するものであり、本研究の毎日計算機環境に関するデータが次々と入ってくる状況に適している。

2 分散計算機環境

一口に分散計算機環境といってもその規模は様々である。計算機が数台しかない環境では環境の管理の必要性はあまり感じられないであろう。また、例えば数十台計算機があってもその利用形態が文書作成や簡単なプログラムの実行だけに限られるときも管理は必要ない。しかし、たとえ数台の計算機であっても、実行するプログラムが計算機の資源を多く費やしてしまうものであったり、各計算機の性能が著しく異なる場合には、管理の必要が出てくる。そこで実際に、管理の必要のある典型的な分散計算機環境を次のように考える。

- 数十台の性能、種類の異なるワークステーション。
- 数台の性能、種類の異なるプリンタ
- 数台の計算機で利用できる複雑なアプリケーション

これらがネットワークで繋がっているものとする。

3 分散計算機資源の管理法

計算機環境の管理の必要性が出てくるのは、利用者がその使用に支障を来すような問題が発生した時である。そこで本システムでは、次の二つの管理機能を有するものとした。

問題解決 計算機環境に何か問題が起きた時にその原因を調べ、その原因を容易に見出すための問題分析。

問題予測 近い将来問題が発生しそうな箇所を指摘する。

システムが問題を持つ場合、その解決の糸口をつかむための手順は以下のとおりである。

1. CPUの負荷を全般的に測定する。
2. メモリの問題を調べる。
3. メモリに問題がない場合、ディスク入出力の問題を調べる。
4. ディスクとメモリに共に問題はないが、システムが障害を起こしている場合、原因はCPUの負荷にある。

単体環境と分散環境の大きな違いは一つのプロセスを実行するにもどのマシンを使うかという選択が可能なことである。たとえCPUが違っていても同様な機能を持つものがあればそれで代用できることも考えられる。また、ユーザの好みなどによって実行可能なマシンが複数あるにもかかわらず特定のマシンに偏りがちである。さらに、計算機のCPUの違いや、OSのバージョンの違いによりその速度や実行可能性はばらばらである。

このため、利用できる計算機が複数になると管理の複雑さは格段に増加する。もともと、計算機のチューニングは計算機に関する深い専門的知識を必要とするものであり、その上計算機の種類まで考慮に入れるとなると人間の手では手に負えなくなる。そこで、このような計算機資源の管理を支援するためのシステムが必要となる。

4 帰納学習を使った管理

帰納学習は知識獲得のボトルネックを解消するための強力な道具の一つである。その帰納学習を使って異なる計算機間の利用頻度の関係などを導くことによって、管理を支援することができる。

このように帰納学習を使って、散環境の管理支援に適用する場合、帰納学習システムに要求される機能は次のとおりである。

1. 数値および記号データを扱える。
2. 学習が高速に行なわれる。
3. 逐次的に学習が可能である。

第1番目の機能は、計算機環境の頻度を示す属性は、ほとんど数値で表され、さらに計算機の名前、利用者の名前などが必要となるため必須である。第2番目の機能は、分析の性質上数千のデータを扱う場合があるため、利用者の使用感を考えると必要な機能である。この場合特徴的なのが第3である。計算機環境の頻度を示すデータは日々増加している。そして、その都度学習をやり直していたのでは時間がかかってしまう。以前の学習結果を利用して学習の方が効率が良いと考えるのは当然である。また、逐次的に学習を行なうことによりデータの変化を調べることが可能になる。

このような要求される機能を持った学習システムとして、逐次型ISL [2] を採用した。

*The development of Incremental Inductive Learning System

¹Yoshiyuki OHKUMA, Hayato OHWADA, Fumio MIZOGUCHI

²Faculty of Sci. and Tech., Science Univ. of Tokyo

5 使用するデータ

計算機環境の分析を行なうためには、各計算機資源について細かい情報を集めなければならない。ここではこれらの情報を OS に備わっている機能を使って収集した。その種類は大きく分けて以下に示す 3 つである。

コマンド 各ユーザが使用したコマンド、CPU やメモリの消費量、実行時間など。

ユーザ ユーザが計算機の使用を開始した時間、使用計算機の名前などを記録する。

計算機 純粋な計算機の処理能力、メモリ、スワップの量など。

ディスク ディスク容量、転送速度、使用消費量など。

プリンタ プリンタの種類、使用量など

これらのデータを定期的に収集し分析に用いる。

6 分析、予測方法

集められたデータを元に分析、予測を行なう。まず、問題分析手順は、次のとおりである。

1. 問題になりそうな属性について、学習を行なっておく。
2. いざ問題が発生した時に、以前の学習結果と現在のデータを照らし合わせ、その範囲外となる属性を探す。
3. その属性が問題の原因であると予想できる。

さらに予測手順は次のようになる。

1. 学習を一定機関毎に行なっておく。
2. 各学習結果を比較し、その変化量を調べる。
3. その変化量から限界到達までの時間を予測する。

7 分析例

ここである特定の計算機が非常に遅くなった場合の問題分析例を示す。まず、あらかじめ計算機の資源を多く要求する特定のプログラムが一日に実行されているマシン名、回数、CPU 使用量、I/O 回数、メモリ使用量を入力として学習を行なわせておく。その入力と学習例を図 1 に示す。

入力

```
epoch: MAC=r10, COMMAND=2, CPU=3.94, IO=51542, MEM=8791
emacs: MAC=usa, COMMAND=67, CPU=3.28, IO=1901, MEM=5208
xdvi: MAC=feb, COMMAND=37, CPU=1.57, IO=20469, MEM=4645
.....
```

学習結果

```
epoch: MAC=r10, COMMAND>=1 & COMMAND<=5 & CPU>=3.25 & ....
emacs: MAC=feb, COMMAND>=15 & COMMAND<87 & CPU>=2.96 & ..
.....
```

図 1: 入力と学習結果

まず、この結果から CPU の消費量の大きいプログラムをチェックする。今回は、find、epoch、emacs、atrm など重いプログラムであることが分かった。

次にこれらのプログラムの 1 カ月前の学習結果と現在の消費量を比較した。すると、現在は、epoch、emacs などの文書作成に関するプログラムが数多く実行されていることが分かった。

さらに、これらの重いプログラムの特定の計算機に集中度を調べるために、1 カ月前までのこれらのプログラムの利用データと最近 1 カ月のデータを別のクラスとして学習を行なった。その結果の一部を図 2 に示す。

学習結果

```
old: MAC=r10, COM=epoch & COMMAND>=5 & COMMAND<=10 & ....
now: MAC=r10, COM=epoch & COMMAND>=11 & COMMAND<=15 & ..
.....
```

図 2: マシン毎の利用データ

この結果から、ある特定の計算機に明らかに利用が集中していることが分かった。そこで、対策として利用者に他の計算機で同プログラムを動かすよう促した。

8 評価

このような、分析実験から分かった帰納学習を使った分析の利点は次のようなものである。

- 異なる計算機での利用データを学習させることによって、一つの計算機を見ていただけでは分からない問題点を発見することができる。
- 逐次的に学習を行なうことによって、データの変化傾向がわかる。
- 複雑な分布を持つ利用データでも、分類規則で分類してくれるので、データの傾向が把握しやすい。

9 まとめ

本稿では、分散計算機環境の問題分析、予測に帰納学習システム、逐次型 ISL を用いた管理支援システムを作成した。帰納学習システムを使うことにより、複雑なデータ間の関係を見出すことが可能となり、複数計算機の管理が容易になった。

参考文献

- [1] ARINZE, B. and IGBARIA, M. A knowledge based decision support system for computer performance management, *Decision Support Systems*, 8 (1992), 501-515.
- [2] 大熊喜之, 大和田勇人, 溝口文雄 数値データの逐次型帰納一般化一般化アルゴリズム, 人工知能学会 第 7 回全国大会 (1993), 143-146.