1D−9

帰納学習ツールISLによるケース分析(2) - 実現 -*

大熊 喜之† 杉本 勉‡ 大和田 勇人† 溝口 文雄†

†東京理科大学 理工学部 ‡NTTデータ通信(株)

1 はじめに

問題解決を行なう時に,人はまずその問題を分析しようとする.これは,問題となる対象から多くのデータを集め,それを細かく分類することによってその傾向を把握し,対象の全体像や特徴を明らかにすることであり,ケース分析という.

一方,知識獲得のボトルネックを解消する方法として活発に研究されている帰納学習は,与えられた事例データから何らかの規則を導き出すものである.これは見方をかえると,事例データをいくつかに分類するものであるといえる.

従来の帰納学習システム FOIL,GOLEM などは基本的に記号データしか扱えないため,問題によっては,数値データを扱う必要のあるケース分析では,数値データを記号データに置き換えるという作業が必要となり,そのプロセスが複雑になってしまう.そこで我々は数値,記号の両データから同様に学習が行なえる帰納学習ツールISL(Induction System for Learning)を開発した.

本稿では,この学習ツールISLをケース分析に用いた場合のその有効性と実現方法について述べる.ここでは,ケース分析の対象としてネットワークに継れた計算機環境の資源の最適利用問題を取りあげた.

2 ケース分析

ケース分析は次の四つのステップに分けられる.

1. **分析データの決定** そのデータを調べれば,問題としている対象が明確化されるというデータを決定する.
2. **分析データの収集** データをできるだけ多く集める
3. **分類** 集まったデータを統計的手法など様々な方法を使って特徴ごとに分類する.
4. **分類結果からのデータの評価** 始めに想定したように問題が明確にならなければ,もう一度 *1* に戻る.

このような四つのステップを繰り返し行なうことによって問題を分析していく.ここで比較的自動化可能なのがステップ *3* である.これをコンピュータに支援させるために帰納学習ツールISLを適用する.

3 帰納学習ツールISL

我々の開発した帰納学習ツールISLはGOLEM [2] のアプローチに沿って,数値も扱えるように拡張したものである.そして記述言語は Quintus Prolog であり,Sparc Station 2 上で動作している.[1].

ISLの学習は,いくつかの属性を持つ事例データに対して一つのクラスに分類されているようなデータが多くある時に,そのクラスに属する事例データが属性のどの範囲になっているかを計算することによってデータを分類するものである.

そしてさらに,次のような特徴を持っている.

- 他のクラスの属性値にかかるまで,範囲を広げる.
- 分類に無関係な属性は,削除する.
- 計算時間が高速である.約1000個のデータを5秒以下で分類する.

これらにより,単に第2節で述べたステップのループを早く行なうことができるようになるだけでなく,分類する属性を変更する,特異なデータは削除する,といったことも簡単にできるようになる.従って,様々な種類の属性をもつケース分析の支援に有効であるといえる.

3.1 帰納学習ツールISLによるケース分析の方法

ISLを使ったケース分析の方法は,第2節の手順に沿ったものだが,分類する時にクラス分けをする必要がある.実際の手順を以下に示す.

1. 数種類の属性を持つデータを用意する.データや属性の数は多いほどよい.
2. 注目する属性を一つ決める.他の属性との関係を調べたい属性を選ぶ.
3. 決めた属性が記号データならばそのままクラスとし,数値データなら適当に範囲を分割してクラス分けをする.
4. 学習を行なわせる.結果は制約式,そして数値データならばその範囲も示した散布図で示される.
5. 結果を見て,その分類に何らかの特徴を見い出せればよい.
6. 結果を見て,うまく分類されていなければ,
   - クラス分けをする属性を変更する.
   - クラス分けの分割位置を変更する.
   - 使用する属性を削除,変更,追加する.

   などを行ない,もう一度学習を行なう.

4 ケース分析の例

本節で,実際にISLを使ったケース分析の例を示す.対象は,ネットワークでつながれた計算機環境の資源の有効利用である.

この場合のケース分析の目的は,計算機の利用頻度を最適化するために,現在の各計算機の使用頻度とユーザーの利用頻度の関係やそれらの特徴を見つけることである.

---

*An application of inductive learning tool, ISL, to knowledge analysis and synthesis

†Yoshiyuki OHKUMA, Hayato OHWADA, Fumio MIZOGUCHI, Science University of Tokyo

‡Tsutomu SUGIMOTO, NTT DATA Comm. Sys. Corp.

data	machine	user	cpu	core	connect	process	login
1992/10/2	bacon	takase	22.00	15,862	764	185	1
1992/10/17	mercury	ebara	14.45	63,586	123	356	2
1992/11/1	venus	katuta	45.48	1,210,880	111	234	1
1992/11/5	bacon	ohkuma	14.00	55,629	392	688	1
....	....	....	....	....	....	....	....

表 1: 入力データの形式

4.1 使用するデータ

入力データとして表 1 のような属性を採用した. 各属性は左から, 日付, 計算機名, ユーザ名,CPU 使用時間 (cpu), 平均メモリ使用量 (core), ログイン時間 (connect), プロセス数 (process), ログイン回数 (login) である. これらのデータは, 各計算機で毎日ユーザごとに収集される.

4.2 実験

まず,1260 個の事例データから, CPU 使用時間と他の属性との関係を調べるために,CPU 使用時間の属性値を最大値 (467) から最小値 (0) までの幅で四等分 (0.0, 116.8, 233.5, 350.3, ∞) して, それぞれ $o1, o2, o3, o4$ という名前でクラス分けをした.
これによる学習結果が次の制約式である.

```
IF [core=(minf,155329)] THEN o1
IF [core=(155329,1272220),connect=(minf,856)]THEN o1
IF [core=(1272220,2694205)] THEN o1

IF [process=(382,inf)] THEN o2

Otherwise o4
```

ここでの問題点は, クラスが o1,o2,o4 の三種類しか現れていないことである. これは,CPU 時間に偏りがあるためでこれを訂正するため, 分割位置を (0, 20, 80, 200, ∞) と変えた.
そして, 次のような結果を得た.

```
IF [core=(minf,35982),connect=(minf,644),
    process=(2,inf)] THEN o1
IF [core=(minf,8723)] THEN o1
IF [core=(15862,66463),process=(minf,452),
    connect= (8,inf)] THEN o1
...
IF [core=(minf,350540),connect=(0,inf),
    login=(minf,13),process=(269,inf)] THEN o2
IF [connect=(131,856),process=(minf,411)] THEN o2
IF [core=(minf,77988)] THEN o2
IF [core=(155329,350540)] THEN o2

IF [connect=(minf,856),process=(335,inf)] THEN o3
IF [core=(minf,415893)] THEN o3

Otherwise o4
```

ここでは, クラスは全て現れているが, 今度はクラス o1 の制約式が 9 つもあり, 多過ぎてわかりにくい. そこで, これらの制約式をよく見てみると, core と connect の属性値が他の属性値に比べて cpu に強く影響していることが読みとれる. そこで, 使用する属性をこの 2 つに絞ってさらに学習を行なわせた.
その結果は, 次のようになった.

```
IF [core=(minf,21662),connect=(minf,644)] THEN o1
IF [core=(21662,35982)] THEN o1
IF [core=(minf,15862),connect=(644,inf)] THEN o1
...
IF [core=(minf,120139)] THEN o2
IF [core=(121405,350540)] THEN o2

IF [connect=(0,856)] THEN o3
IF [core=(minf,415893)] THEN o3

Otherwise o4
```

属性が削減され, 制約式も少くなった段階で散布図と制約式を重ね合わせた図を表示させた. それを図 1 に示す.

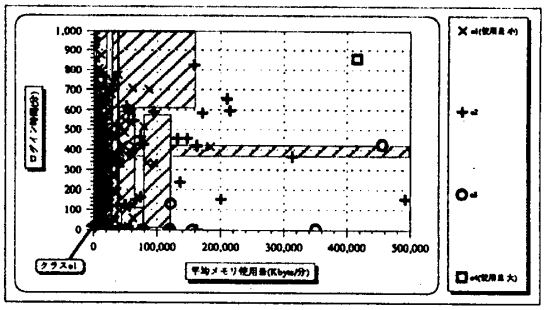

図 1: CPU 時間による分類と平均メモリ使用量と接続時間との関係

この図 1 から CPU 時間を多く使用しているユーザはメモリも多く使う傾向が見られるが, 接続時間はほとんど関係がない. これは学習結果のクラス o2,o3 でほとんど属性 connect が削除されていることからも明らかである.
結果的に, 属性の数が 5 つから 3 つに減り, 事例数 1260 個のデータが 14 個の制約式で分類できた.

4.3 評価

実際に ISL を使ってケース分析を行なうことによって次のようなことが可能であることがわかった.

- クラスの分割法の変更.
- 不必要な属性の自動または, 半自動的な削除.
- 学習結果を散布図と合わせて表示させることによる, 属性間の関係の明確化.

5 まとめ

帰納学習をケース分析に適用することの有効性を LAN 管理支援を例にとって示した. さらに, 属性やデータを多く含むようなケース分析に対して帰納学習ツール ISL が有効であることを示した.
我々は現在までに, 帰納学習ツール ISL を SIS システムの開発支援 [3], そして LAN 環境の二つに適用している. 今後の展望として, 他の環境への適用の可能性の調査の他,ISL 自体の機能の充実, 例えば, クラス分けの自動化などを実現したい.

参考文献

[1] 大和田他:多ソート帰納学習システムの設計 (I) - 一般化アルゴリズム -, 情報処理学会第 46 回全国大会予稿集,1993.

[2] S. Muggleton. Inductive logic programming. In *New Generation Computing*, 1991, Vol 8, pages 42-61, 1990.

[3] 大和田 他:帰納学習をとり入れた意思決定支援システムの設計, 情報処理学会第 45 回全国大会,4H-1,1992.