オブジェクト指向型決定木学習システムの開発

3AE−10

増田 剛* 坂本 憲広** 牛島 和夫*

* 九州大学大学院システム情報科学研究科 ** 九州大学医学部附属病院医療情報部

1. はじめに

データ収集、蓄積技術の進歩に伴い、様々な分野で大規模なデータベースが構築されている。データベースからの知識発見 (KDD)[1] は、このような大量のデータを解析し、人間によって有用な知識を得るための技術である。KDD 処理は、データの精錬、前処理から、データ発掘、発見されたパターンの解釈までの多段階の処理から構成される。機械学習の分野で研究されている決定木学習は、この KDD 処理におけるデータ発掘部分に適用することのできるシステムの 1 つである。代表的なものに C4.5[5] がある。

KDD 処理は反復的であり、ユーザとの相互作用的な処理を伴う。なぜなら、あるデータに対してどのような手法が有効であるかという明確な指標が存在しないからである。そのためユーザは、適用する手法を状況に応じて調節あるいは変更しながら、繰り返し KDD 処理を行なう必要がある。それゆえ、決定木学習システムをこのような KDD 処理の一部として用いる場合には、手法の変更や調節が容易に行なえるような高い柔軟性が必要となる。しかし、既存の決定木学習システムでは、柔軟性に重点を置いていないため、システムの一部分の変更や拡張はユーザにとって困難である。

そこで本研究では、柔軟性に重点を置いた決定木学習システムを開発する。システムの柔軟性を高めるために、オブジェクト指向技術を用いて学習システムの各構成要素を小さなオブジェクトに分解し再構成する。この際、デザインパターンという設計法を用いることにより、より高い柔軟性を実現する。

2. デザインパターン

デザインパターンとは、オブジェクト指向ソフトウェア設計において頻繁に現われる重要な設計に名前を付け、体系化したものである。1 つのデザインパターンは、パターン名、目的、適用可能性、結果といった項目から構成される。これらのパターンを組み合わせることによって、柔軟性の高いシステムを容易に構築することができる。

本研究では、デザインパターンとして、Gamma らが提案したデザインパターンカタログ [2] を使用する。これは、オブジェクトの生成、構造、振る舞いに関する 23 個のパターンを規定したもので、パターンの適用可能性や適用結果を明確かつ実戦的に記述している。以下では同カタログのデザインパターン名を断りなしに用いる。

Development of An Object Oriented Decision Tree Learning System.

Go Masuda*, Norihiro Sakamoto** and Kazuo Ushijima*

*Graduate School of Information Science and Electrical Engineering, Kyushu University.

**Department of Medical Informatics, Kyushu University Hospital.

3. 設計と実装

3.1 システムのホットスポット

一般に柔軟性の高いシステムを構築するためには、システム中で将来変更や拡張が予測される部分を見極めることが重要である。本研究では文献 [4] に従いこれをホットスポットと呼ぶ。そして、学習システムに対する要求分析と、種々の学習システムの調査、さらには試作したプロトタイプに基づき、決定木学習システムにおいて以下の 8 個のホットスポットを同定した。

データセットの入力及び生成: 学習に用いる事例として、様々な形式の事例を柔軟に扱いたい。またその入力方式についても柔軟性を持たせたい。

属性の種類: 事例を構成する属性として、任意の型の属性を考慮したい。

決定木の構造: 決定木はシステムの中心となるオブジェクトであるため、決定木に対する操作はシステム全体に影響を及ぼす。そのため、決定木操作の安全性を高めたい。また、決定木の構成要素に依存せずにそのような操作を実現したい。

テストの種類: 事例を分割するための属性に関するテストの種類に柔軟性を持たせたい。

テストの選択法: 様々な選択法を属性の種類に依存せずに扱いたい。

ノイズデータの処理: テストの選択法や属性の種類に依存することなしに、様々なノイズデータの処理法を扱いたい。

決定木の枝刈り: 様々な枝刈り手法を柔軟に扱いたい。

決定木の評価: 様々な観点から学習結果を評価できるように、様々な評価法を柔軟に扱いたい。

3.2 設計

前節で示した各ホットスポットについて、要求される機能を明確にし各部分に適用するデザインパターンを決定した。

データセットの入力及び生成部では、任意の型の属性を扱うという要求から、データセットの生成が属性の種類に依存してはならない。そこで、ここに *Abstract Factory* パターンを適用する。これは属性オブジェクトの生成過程を隠蔽するためのパターンである。

また、事例を分割するテストの選択法については、様々なテスト選択手法に共通する部分と共通でない部分を分離し、必要最小限の変更で新しい選択手法を導入できることが要求される。ここには *Template Method* パターンを適用する。これはアルゴリズムの一部分の変更を可能にするパターンである。

このように、各ホットスポットに対してそこに要求される機能を満たすデザインパターンを選択する。最終的に本システムでは、8 個のホットスポットに対し、延べ 15 パターンを適用した。

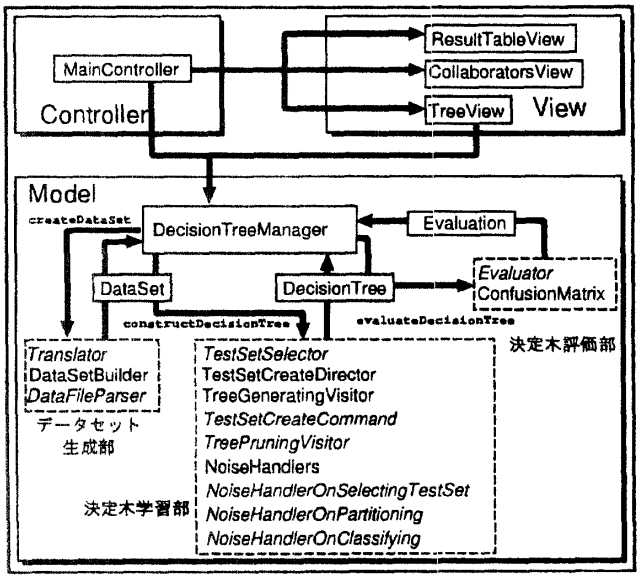

図 1: システムの構成図

3.3 システムの構成

以上の設計を基に、オブジェクト指向言語 Smalltalk を用いて、決定木学習システムを実現した。システムの構成図を図 1に示す。システムは、モデル、ビュー、コントローラの三つの部分から構成される。モデルは学習システムの中心部分である。ビューは表示を司る。コントローラはユーザからの入力を管理する部分である。さらに、モデル部分は、学習に必要なデータセットの生成部、データセットから決定木を導出し、また、決定木の枝刈りなどを行なう決定木学習部、決定木の評価を行なう決定木評価部から構成される。各部分には、関係する一連のクラス群が含まれる。斜体で表わしているクラス名は、そのクラスが抽象クラスであることを示す。ユーザが実際に自分自身のデータに対して学習システムを適用する場合には、これらの抽象クラスから派生したサブクラスを導入することで実現することができる。

4. 評価

本システムでは、システムの柔軟性を得るためにプログラムの効率よりも構造を重視しているため、学習に要する時間が大きくなることが予測される。そこで本節では、実装したシステムを実行効率の観点から評価する。

実験では、デザインパターンを用いて実現した本学習システム (デザインパターン有) と、デザインパターンを用いずに試作したシステム (デザインパターン無) について、決定木の生成と枝刈りにかかる時間を 5 回測定し、その平均をとる。実験に用いるデータセットは、カルフォルニア大学アーバイン校において管理されている機械学習のためのデータライブラリを利用する。測定結果を表 1に示す。

結果より、デザインパターンを用いた本システムは、その実行効率がデザインパターンを用いていないシステムに比べ、平均約 2 倍に悪化していることがわかる。これは、デザインパターンを用いた場合、継承よりもオブジェクトの合成を多用するということ、そして各オブジェクトのカプセル化の度合がより高いという理由から

表 1: 測定結果: 学習に要する時間

	crx	hepatitis	iris
デザインパターン有	367.1	62.9	6.3
デザインパターン無	123.1	26.1	2.9
	labor-neg	soybean	vote
デザインパターン有	2.1	144.7	4.0
デザインパターン無	1.0	112.9	1.8

単位 [秒]

IBM PC360 CPU: Pentium Pro 150MH, メモリ: 64MB

ら、通常のオブジェクト指向プログラムよりもオブジェクト間のメッセージの送信数が増加することが原因であると推定している。

システムの柔軟性と実行効率はトレードオフの関係にあるので、実際の大規模データベースを解析する場合を考えると、今後検討すべき問題である。

5. 議論

本システム (デザインパターン有) では、学習の対象となる事例をオブジェクトとして扱い、事例が任意の型の属性を持つことができるように、学習アルゴリズムを拡張している [3]。これにより、応用分野ごとに特有の構造を柔軟に扱うことができる。さらに、任意の型の属性に対して、事例を分割するための任意のテストを導入することが可能である。既存の学習システムにおいて、このような属性やテストの種類は、学習アルゴリズムと密接な関わりを持つため、新しい属性やテストが増える度に学習アルゴリズムを修正しなければならず、それらの変更や拡張が困難である。しかし、本システムにおいては、デザインパターンを用いて属性やテストの種類と学習アルゴリズムを完全に分離しているために、ユーザは既存クラスのサブクラスとして新しい属性やテストを導入し、必要最小限のコードを追加するだけでよい。

6. おわりに

本研究では、KDD 処理のような実際の応用で使用できる柔軟な決定木学習システムの開発について述べた。KDD が成功するためには、ある単独の技術だけで解決するのではなく、種々の技術を組み合わせ、統合していくことが重要である。そこで、今後は、KDD 処理全体に視野を広げデータの獲得、精錬からデータの解釈、視覚化までを継目なく柔軟に扱うことのできるメタな枠組について検討する予定である。

参考文献

[1] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, H.: "Advances in Knowledge discovery and data mining," AAAI/MIT Press, 1996.

[2] Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: "オブジェクト指向における再利用のためのデザインパターン," ソフトバンク株式会社, 1995.

[3] 増田、坂本、牛島:"概念学習システムにおける事例のモデル化についての検討, " 第 53 回情報処理学会全国大会, 1996.

[4] Pree, W. "デザインパターンプログラミング," 株式会社トッパン, 1996.

[5] Quinlan, J. R.: "*C4.5: Programs for Machine Learning*," Morgan Kaufmann Publishers, San Mateo, CA, 1993.