帰納論理プログラミングと関係データベースの統合処理システムの設計

1T－1

大塚勇介　大和田勇人　溝口文雄

東京理科大学 理工学部

1. はじめに

データベースからの知識発見では、データマイニングアルゴリズムそのものの性能だけでなく、アルゴリズムとデータベース管理システムのインタラクションや統合も重要な側面となる。

本稿では、帰納論理プログラミング(ILP)のプロセスを関係データベース管理システム(RDBMS)上で実現する方法を述べ、これらを統合した推論システムを提案する。これにより、RDBMS の高度な検索・演算機能を ILP に取り込むことができるだけでなく、データベース操作言語(SQL)により ILP の処理を補い拡張することが可能となる。

2. 関係代数と関係論理

関係データベースと ILP には密接なつながりがあることは知られている。この関連を図 1 に示す。関係データベースは集合論からなる関係代数と、述語論理に基礎をおく関係論理により理論化された。これにより、ILP の仮説言語である一階述語論理をデータベース上で表現し、SQL を用いて問い合わせることが可能となる。本システムでは、前処理でこのマッピングを行い、ILP の述語形式をテーブル形式へ変換する。

データベース	述語論理
relation name p	predicate symbol p
attribute of relation p	argument of predicate p
tuple $< a_1, \ldots, a_n >$	ground fact p $(a_1, \ldots, a_n)$
relation p : a set of tuples	predicate p : defined extensionally by a set of ground facts
relation q : defined as a view	predicate q : defined intensionally by a set of rules or clauses

図 1 関係データベースと述語論理の関連

3. RDBMS による仮説のカバーリング

本システムでは、ILP の推論エンジン部で構築された仮説をデータベースに対して問い合わせることにより、その仮説が有意かどうかを検証する。仮説の有意性は、仮説が包含する事例数とその仮説の特殊化率（例外的事例を含む割合）から判断する。

しかしながら、仮説が包含する事例数を求めるために、仮説を構成する述語レベル、すわなちテーブルごとに検索していたのでは、処理速度やコストの点で問題が大きい。そこで、仮説そのものを全く等価な意味を持つ SQL クエリーへ変換することにより、仮説が包含する事例数を単一クエリーで直接的に得ることができる。

例えば、動物(animal world)に関する推論問題においてある仮説が生成された場合、述語論理と SQL により、それぞれ以下のような同等の記述が可能である。

＜仮説表現＞

・**意味**：動物 A が羽を持つならば A のクラスは鳥類である

・**節形式**：class(A, bird) :－ has_covering(A, feathers) .

・**SQL**：
```
SELECT count(*)
FROM   class table1,  has_covering table2
WHERE  table1.class = 'bird' AND
       table2.covering = 'feathers' AND
       table1.animal = table2.animal ;
```

また上述の仮説が RDBMS によって、実際にどのように処理されるのかを図 2 に示す。

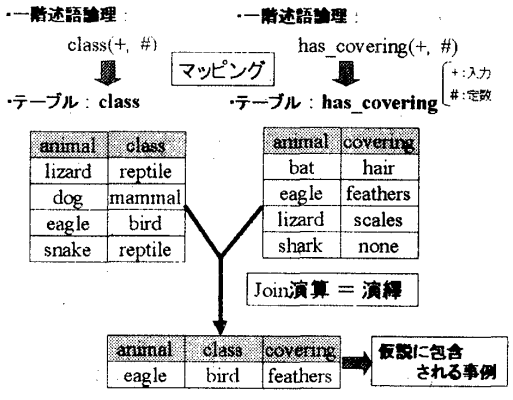

図 2　RDBMS による仮説のカバーリング

前述したようにデータベース内のデータ構造を述語に基づいた形式で定義しておくことにより、節形式をテー

Yusuke OHTSUKA, Hayato OHWADA, Fumio MIZOGUCHI
Faculty of Sci. and Tech. Science University of Tokyo

ブルのリレーションで表現することが可能となる。仮説を表す節形式文の実行は、SQL の結合演算(join)による演繹処理として実行される。述語（テーブル）間の関係を、仮説の変数にあたる属性で結合し、結果リレーションと事例数を得る。ILP 推論部では、この仮説が包含する事例数を仮説の有意性を判定するために利用する。

4. ILP と RDBMS の統合処理システム

本システムにおける推論エンジンである ILP は、本研究室で開発された GKS(GaKuShu)をベースに実装した。ILP の目的は学習ターゲットである正事例とその背景知識から、できるだけ多くの正事例を包含し、負事例を含まない、あるいは十分に特殊な仮説を発見することにある。GKS は A*アルゴリズムによりトップダウンに仮説空間を探索する。

本システムでは、推論エンジン部は主に述語定義・仮説生成プロセスを実行し、実際の仮説のカバーリングは RDBMS が行うというように、ILP と RDBMS との協調処理により帰納推論を実行する。

4.1. システム構成

本システムの構成を図 3 に示す。本システムの ILP 推論エンジン GKS は、Java により実装した。推論エンジンと RDBMS とのインターフェースには JDBC(Java Database Connectivity)を用いている。本システムは Java と SQL により実現されており、OS などの環境に依存しない。また、RDBMS には Oracle や Microsoft Access などの標準的な関係データベースを利用可能であり、移植性に富む。

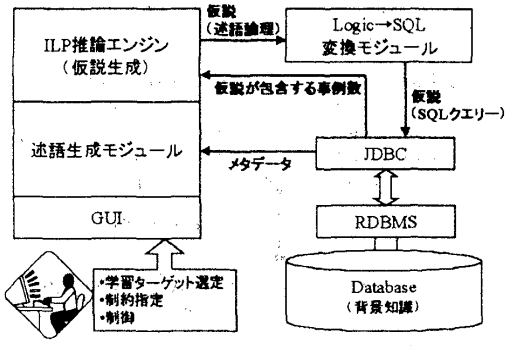

図3　システム構成

本システムにおける推論プロセスを以下に述べる。まず、ユーザーはデータベースメタデータを利用して、データベースの構造や属性に関する情報を取得し、正事例や負事例・背景知識など、推論で用いる述語を定義する。そして、定義した述語形式にデータベース内のテーブル構造をマッピングする（述語生成モジュール）。同時に、ユーザーはILPの仮説空間を制御するためのバイアス、特殊化率などを指定する。

次に、定義された背景知識を利用して仮説を生成していく（推論エンジン）。このプロセスは完全な帰納論理プログラミングである。ここで生成された述語論理ベースの仮説は、RDBMS で処理できるように SQL 形式へと変換される（仮説変換モジュール）。RDBMS は SQL により表現された仮説が背景知識内の各事実を満たすかどうかを検証し、仮説に包含される事例数を返す。推論部ではこの事例数を利用して仮説の一般性と特殊性を計算する。制約条件を満たした段階で仮説はルールとなりユーザーに提示される。ユーザー指定の制約を満たすような全てのルールが発見されるまでこの処理が続けられる。

4.2. 適用

本システムを電子メールの分類問題に適用した。得られるルールは推論部の ILP によるものと同等のものである。本システムによる利点は、ILP と RDBMS を統合することにより、データベースに蓄積されたデータを容易にマイニングすることができ、得られたルールをダイレクトにデータに適用できることである。また、インデキシングによる高速検索、演算処理の最適化、堅牢なデータ管理機能など、データベースの高度な処理機能を推論エンジンに組み込み、利用することが可能となる。

5. おわりに

本研究ではデータベースからの知識発見プロセスに対する ILP の適用を、推論システムと関係データベースの統合という観点から行った。しかし、データベースという二次記憶を利用しているために、メモリ上で行う ILP と比較して、処理時間がかかることは否めない。本システムは、データ構造をテーブル形式へマッピングすることにより実現しているが、推論部のアルゴリズムを RDBMS の処理機構に最適化したものに設計することが不可欠である。今後、提案・実装していく予定である。

参考文献

[1] Hendrik Blockeel and Luc De Raedt : Relational Knowledge Discovery in Databases, Inductive Logic Programming 6th International Workshop ILP-96, 1996.