2M－1

CAMLET: 帰納的学習システム構築支援環境 (1)
－仕様と実装の対応付け－

根岸 直矢　　酢山 明弘　　山口 高平

静岡大学

1　はじめに

近年, 知識工学においては, 問題解決用部品をライブラリー化し, それらを再利用して, 知識システムを開発する方法論の整備が進んでいる [Heijst, 1995]. 一方, 帰納的学習の研究においては, 決定木や分類システムなどの帰納的学習システムが広く普及してきたが, システムの性能は, データセットに依存して大きく変化し, 与えられたデータセットに対して有用な帰納的学習システムの仕様については, 試行錯誤的に同定し, 多くのコストがかかっているのが現状である.

以上の背景から, 本稿では, 帰納的学習システムの構成部品を同定し, それらを組み合わせることにより, 所与のデータセットに対して有用な帰納的学習システムの仕様とコードを自動的に生成する環境である CAMLET を提案するとともに, 実験を通して評価を行う.

2　CAMLET の概要

帰納的学習システムを自動的に構築するには, 部品ライブラリー (プロセスオントロジー) と部品で処理されるデータの型 (オブジェクトオントロジー) を事前に体系化する必要がある. 以下, オントロジーとシステムの基本設計について説明する.

2.1　帰納的学習システムのオントロジー

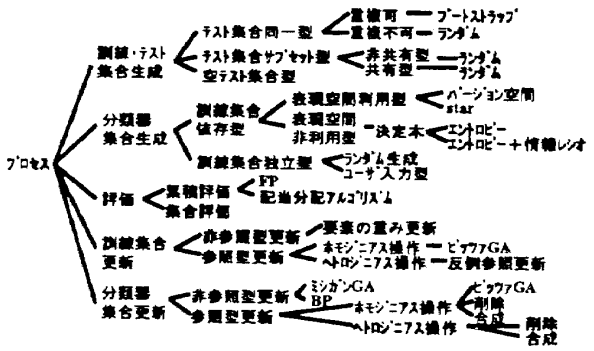

図 1: プロセスオントロジーの概念階層

代表的な帰納的学習システムを分析した結果, 帰納的学習システムは, 訓練・テスト集合生成, 集合訓練更新, 分類器集合生成, 分類器集合更新, 評価の５つのプロセスが組み合わされて構成されていることが判明した. そこでこれら５つを最上位階層に位置づけている. また, 概念階層のリーフが実際にコードレベルで用いられるプロセスである. このようにして作成した概念階層を図１に示す. オブジェクト階層はプロセスによって操作されるデータ構造に着目して階層化をおこなっている.

2.2　CAMLET の基本設計

CAMLET は, 問題解決用部品を利用した知識システム構築法に基づいて設計されている.

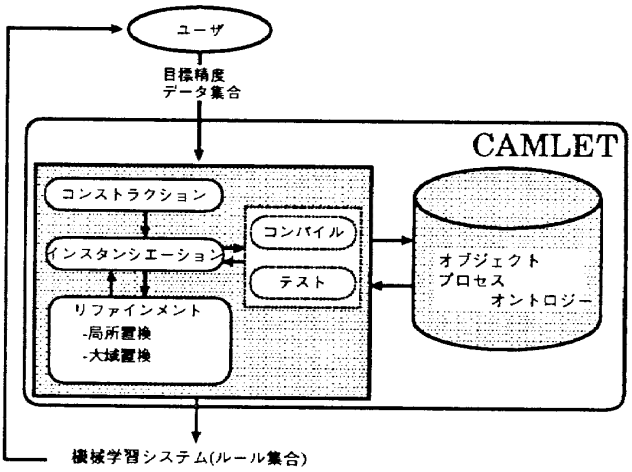

図 2: 帰納的学習システム構築法

まず, ユーザは CAMLET にデータセットと構築する帰納的学習システムに要求する目標精度を入力として与える. コンストラクションではプロセス階層の上位プロセスを用いて制御構造を作り, 次にそのプロセスを制約情報を利用してリーフにあたるプロセスに置き換え初期仕様を作成する. インスタンシエーションでは, 与えられたデータ集合のデータ型を使って初期仕様として選択された末端プロセスの input, output を決定する. こうして得られた仕様からコード生成を行い実際に実行可能なシステムにしテストを行い, その結果がユーザからの要求を満たしていればそのシステムを出力し終了する. 要求が満たされない間は, リファインメントによってシステムの仕様の変更が繰り返される. リファインメントではプロセスを別のプロセスに変更する局所置換とシステムの制御構造自体を変更する大域置換がある.

CAMLET: A Commputer Aided Inductive Learning Systems Engineering Environment
- Mapping Specications into Implementation Details -
Naoya Negishi, Aihiro Suyama, Takahira Yamaguti
Shizuoka University

3　仕様と実装の対応

仕様からプログラムコードを自動生成する場合，プロセスの結合時にプロセス毎に入出力仕様が異なるためデータの型の変換を行う必要がある．CAMLETが帰納的学習システムを構築する時に用いられるプロセスの中で入出力仕様が異なるプロセスは分類器集合生成と分類器集合更新であり，そのデータの型としては決定木，ネットワーク構造，if-thenルールの3つがある．これらの3つのデータの型を図3に示すように分類器へと変換するデータの型変換プロセスを作成し，必要に応じて利用することで任意のプロセス間で入出力仕様を保持することが可能である．例えば，決定木のデータの型変換では，ルートからリーフまでのパスに対して分類器を生成する．すなわち、パス中のノードの属性とその値から条件部を作り，リーフの分類クラスから結論部を作る．これをすべてのパスについて行うことで分類器集合を得る．また，分類器はルールの有効性を示す強度を持ち，強度はすべての分類器について同じ値を設定することにしている．

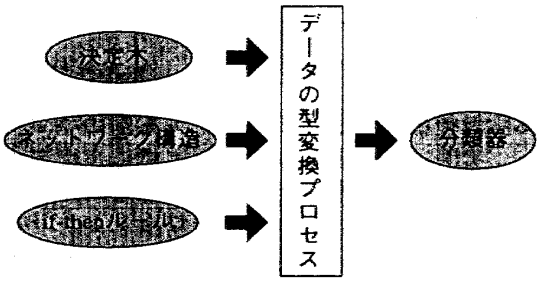

図 3: データの型変換プロセス

4　実験と評価

UCIデータからZooligical Database, Soybean Problemを利用し，14個の部品プロセスに限定して，UNIXプラットフォーム上で帰納的学習システム構築実験を行った．

表 1: 実験データ

データ名	属性数	訓練データ数	テストデータ数
Zoological	18	56	45
Soybean	35	307	376

図4は，与えられたデータ集合に対しCAMLETが構築した帰納的学習システムの性能の変化を示す．訓練集合は，帰納的学習システムを構築する時に用いられ，テスト集合はCAMLETによって構築された帰納的学習システムを評価する時に用いられる．ただし，データセットが小規模であるため，訓練集合とテスト集合は同一のものを用いることにする．また，図4は初期システムからユーザの要求を満たす最終システムまでの性能変化を示し，目標精度は，Zoological databaseで100%, Soybean Problemで90%と設定した．Zoological, Soybeanともに1回だけ大域置換を行いZoologicalでは7回，Soybeanでは4回のリファインメントにより目標精度に到達した．

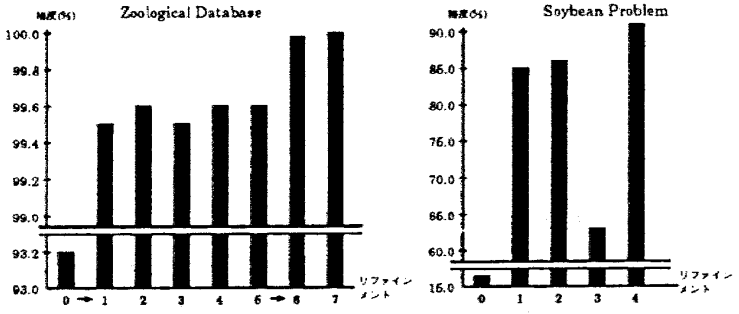

図 4: 構築された帰納的学習システムの性能変化

また，仕様からのコード生成を手動で行った時と自動で行った時に要した時間の比較を行った．手動とは，構築に利用される部品はライブラリーとして用意されておりデータの型変換と仕様を基に部品を使って組み立てる部分を手作業で行った場合であり，自動とはそれら全てを自動で行った場合である．

表 2: 実験時間の比較

データ名	手動 (s)	自動 (s)
Zoological	660	40
Soybean	840	48

仕様からのコード生成が自動的に行えたのは，データの型変換プロセスが対象としているのは3つのデータの型だけであり，決定木だとノードには属性とその値が，リーフには分類クラスが必ず対応しているという限定されたものであるため，データの型変換プロセスによって入出力仕様の保持を行うことが比較的容易に行えたからと考えられる．また，Soybeanでは目標精度が90%であったので4回のリファインメントで終了したが目標精度をさらに高くして実験を行うとより多くのリファインメントが行われることが予想され，その度にコード生成が行われるのでコード生成の自動化の効果がより大きく現れると考えられる．

5　おわりに

データの型変換プロセスを作成し，それによって入出力仕様が保持され，コードレベルでも帰納的学習システムの自動生成が実現できることを確認した．帰納的学習システムの構成空間は巨大であるため，今後，並列処理を導入した探索技法の確立が必要であると考えられる．

参考文献

[Heijst, 1995] Gertjan van Heijst: "The Role of Ontologies in Knowledge Engineering", Dr Thesis, University of Amsterdam, 1995.