小特集 「最近の機械学習」

効率化学習

Speedup Learning

山田 誠二*
Seiji Yamada

* 大阪大学産業科学研究所
ISIR, Osaka University.

1994年8月8日 受理

Keywords: machine learning, search control knowledge, macro-operator, utility problem.

1. 効率化学習とは

一般に，機械学習は，帰納学習，演繹学習，類推，発見学習に分類されるが，これらは学習方法から見た分類である．これとは別に学習の目的からの分類も考えられる．そのような視点での分類の一つが，効率化学習(speedup learning)である．効率化学習は，メタ知識を学習することにより，問題解決の効率向上を実現する機械学習を意味し，制御知識の学習やマクロオペレータ学習などの分野の総称として最近使われ始めている．効率化学習は，概略的に図1のようなシステム構成を持つ．下位システムとして，問題に対し適用可能な操作の集合であるオブジェクト知識のみを持ち，その知識をいかに使うか(例えば，複数の操作が適用可能なときにどれを選択するか)というメタ知識を持たない問題解決システムがある．ここで，問題解決システムは，すべての可能な操作の適用により，解を含む問題空間を生成できるため，原理的には単独でも問題を解ける．ただし，メタ知識を持たないため，膨大な問題空間を効率良く探索することができず，一般にその効率は非常に悪い．

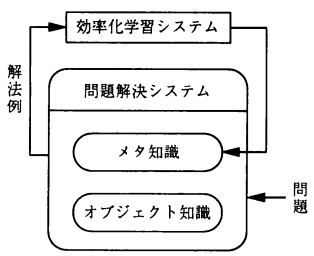

図1 効率化学習の構成

例えば，将棋のコマの動かし方(オブジェクト知識)を知っていても，どのように効果的な手を選択するかというメタ知識を知っていなければ，勝負に勝てないし，また，式の変形を知っていても，解に近づくようにうまく変形を選択しないと方程式を解くことができない．よって，現実的には，オブジェクト知識だけで問題を解くことは困難であり，メタ知識が必須となる．このメタ知識を機械学習により獲得し，問題解決の効率向上を目指すのが，効率化学習である．

主な効率化学習には，探索制御知識(search control knowledge)の学習，マクロオペレータ学習(macro-operator learning)があるが，いずれの学習システムもそのほとんどが，学習手法として，説明に基づく学習：EBL(Explanation - Based Learning)[DeJong 86, Mitchell 86]を用いている．訓練例として，解法例が人間あるいは問題解決システムから与えられる．

以降では，2章でまず説明に基づく学習を簡潔に説明してから，上記の二つの学習のサーベイを行う．そして，3章で，効率化学習の重要課題である効用問題について触れ，4章で応用例を紹介する．

2. 方 法 論

2・1 説明に基づく学習

従来，機械学習において広く研究されてきた帰納学習(inductive learning)[Michalski 83]では，学習させたい目標概念の正例と負例を大量に学習システムに与え，学習システムはあらかじめ与えられているルールを用いて概念記述を一般化・特殊化して，正例を含み，

負例を含まない目標概念の内包的記述を探索する．一般に可能な概念記述の空間は非常に大きいために，帰納学習は多数の訓練例を必要とする．

これに対し，一つの訓練例(正例)に対し，それがなぜ目標概念の例であるのかを領域理論(domain theory)を用いて説明(証明)し，その説明が成り立つ範囲で一般化を行うのが，説明に基づく一般化：EBG(Explanation-Based Generalization)である．これにより，帰納学習における目標概念と関連のある特徴の選択や過度の一般化(over-generalization)などの問題が解決され，非常に効率良く目標概念を学習することが可能になる．また，EBG を用いた学習を，説明に基づく学習：EBL(Explanation-Based Learning)[DeJong 86, Mitchell 86]と呼ぶ．

EBG のホーン節で表現された入出力，処理を以下に示す．

〈入力〉

- 目標概念(goal concept)：学習すべき目標概念を頭として持つ規則節(操作性基準は満たしていない)．
- 訓練例(training example)：目標概念の正例である事実節の集合．
- 領域理論(domain theory)：説明の生成に十分な知識(規則節の集合)．
- 操作性基準(operationality criterion)：一般化のために説明の部分構造を取り出すときの基準．取り出す部分木の葉になるべき述語の集合で与えられる．

〈出力〉

- 操作可能な概念記述：操作性基準を満たす目標概念記述．

〈手続き〉

(1) **説明の生成**：目標概念を領域理論と訓練例を用いて証明する．この証明木を説明と呼ぶ．

(2) **一般化**：まず，説明から操作性基準を満たす述語を根とする部分木を取り除く．次に，残った部分木の生成に用いられた領域理論を用いて，目標概念から後向きに説明を再構成し，規則節を構成している述語の引数間の単一化(unification)を行う．

(3) **マクロ化**：一般化された説明構造の葉の述語，つまり操作性基準を満たす述語の連言(conjunction)として，概念記述の本体を得る．これが出力となる．途中の中間仮説は，葉がすべて真であれば領域理論より必ず証明されるため，概念記述には必要ないので取り除かれる．

なお，EBG(EBL)の詳細は，[DeJong 86, Mitchell 86, 沼尾 88, 山田 91]を参照されたい．

2・2 探索制御知識の学習

前述のように，オブジェクト知識だけで問題解決をする際，一つの問題状態に対し，数多くのオブジェクト知識が適用可能な場合が頻繁に起こり，そこで適切な適用を選択しないと解にたどり着けない．このような選択に用いられる知識は，探索制御知識と呼ばれる[Minton 88]．この探索制御知識を学習することは，問題解決を大きく効率化することになる．ここでは，有名な学習システムである PRODIGY における探索制御知識について見ていこう．

PRODIGY[Minton 88]は，S. Minton らにより開発された学習システムで，説明に基づく学習により問題解決の制御知識の学習を行う．問題解決器として STRIPS-like なプラニングシステムを持つが，その問題解決において，展開すべきノード，サブゴール，オペレータ，オペレータの変数の束縛の四つについて，候補が多数あるため，それらから適切なものを選択しなければ解にたどり着けない．そので，PRODIGY は，これらの選択において，ある特定の候補を選択，拒否，選好する探索制御知識を学習する．それぞれの規則を，選択規則，拒否規則，選好規則という．選択規則と選好規則の例とその意味を図 2 に示す．

PRODIGY の EBL で扱われる目標概念とそれから学習される制御知識を以下に示す．目標概念は，制御のクラスを意味する．

- 成功：成功裏に終わる制御．選好規則が学習される．
- 失敗：その制御と無矛盾な解がないような制御．拒否規則が学習される．
- 単独候補：他のすべての候補が失敗するような制御．選択規則が学習される．

(SELECT OPERATOR (UNSTACK *x y*))
if (and (CURRENT-NODE *node*)
(CURRENT-NODE *node* (HOLDING *x*))
(CANDIDATE-OPERATOR *node* (UNSTACK *x y*))
(KNOWN node (NOT (ONTABLE *x*))))
現在の目標が，(HOLDING *x*) であり，*x* がテーブルの上にないとき，UNSTACK を選択するという戦略を意味する．(KNOWN *node exp*)は，ノードnodeの状態が*exp*とマッチするとき真．

(PREFER GOAL (ON *x y*) OVER (ON *w x*))
if (and (CURRENT-NODE *node*)
(CURRENT-GOAL *node* (ON *x y*))
(CURRENT-GOAL *node* (ON *w x*))
目標の候補として，(ON *w x*)と(ON *x y*)があるとき，(ON *x y*) を(ON *w x*) より先に達成する方が好ましいという戦略を意味する．

図 2 探索制御規則

・目標の相互作用：一度達成された目標が，再び達成されるような制御．選好規則が学習される．

ある制御がなぜ成功(失敗)したのかを説明するために，PRODIGY は，問題の領域レベルの知識だけではなく，問題解決器自体に関する領域理論も用いる．両者の例を図3に示す．この例では，領域レベルの知識は，オペレータの記述である．説明の生成過程は，目標概念を特殊化するという意味で，PRODIGY では特に，説明に基づく特殊化 EBS(Explanation-Based Specialization)と呼ばれている．その結果得られた説明に，テンプレートを適用して，制御知識が得られる．

成功からの学習の具体例[Minton 87]で，学習過程を説明しよう．今，次のような目標概念と訓練例があるとする．訓練例は，目標状態が(HOLDING B)の問題のプラン中で最後に適用されたオペレータ(UNSTACK B)とする．

目標概念 G：(OPERATOR-SUCCESS *op goal node*)
訓練例 E：(OPERATOR-SUCCESS (UNSTACK B)
(HOLDING B) Node 11)

まず，図3の *Schema-S*1，*D*1，*D*2 を後向きに適用していくことにより，G が以下のように展開される．

Specialize (OPERATOR-SUCCESS *op goal node*)；
(AND (AND (MATCHES *op* (UNSTACK *x y*)
(MATCHES *goal* (HOLDING *x*))
(AND (MATCHES *op* (UNSTACK *u v*))
(KNOWN *node* (AND (CLEAR *u*)
(ON *u v*)
(ARMEMPTY)))

簡略化を施して，以下の概念記述が得られる．

(OPERATOR-SUCCESS *op goal node*) if
(AND (MATCHES *op* (UNSTACK *x y*)
(MATCHES *goal* (HOLDING *x*))
(KNOWN *node* (AND (CLEAR *u*)
(ON *u v*)
(ARMEMPTY)))

■問題解決レベルの知識

*Schema-S*1: 直接目標を満たすオペレータは，成功する．
(OPERATOR-SUCCEEDSS *op goal node*) if
(AND (MACHES-EFFECT *goal op*)
(APPLICABLE *op node*)

■領域レベルの知識

*Schema-D*1: UNSTACKの結果は，HOLDINGである．
(ADDED-BY-OPERATOR *goal op*) if
(AND (MATCHES *op* (UNSTACK *b*1 *b*2))
(MATCHES *goal* (HOLDING *b*1)))

*Schema-D*2: UNSTACKの適用条件の記述．
(APPLICABLE *op node*) if
(AND (MATCHES *op* (UNSTACK *b*1 *b*2))
(KNOWN *node* (AND (CLEAR *b*1)
(ON *b*1 *b*2) (ARMEMPTY))))

図3 領域理論

(ARMEMPTY)))

この制御規則は，オペレータが(UNSTACK x y)で，目標状態が(HOLDING x)であり，かつ HOLDING の条件が成り立つとき，そのノードは成功するという弱い規則である．この概念記述と OPERATOR-SUCCESS のテンプレートから，探索制御規則が生成される．成功から，ほかにも上記の例よりも強力な規則が，より前に適用されたオペレータの条件の伝播などにより得られる[Minton 88]．

また，PRODIGY は，積木の世界およびロボットのプラニング，スケジューリングにおいて体系的な実験が行われている[Minton 88]．そこでは，PRODIGY，人間がコーディングした制御知識を用いたシステム，そして制御知識なしのシステムの三つが同じ問題を解いたときの評価がされており，PRODIGY は概ね良好な結果を得ている．

2・3 マクロオペレータ学習

マクロオペレータ (macro operator) とは，オブジェクト知識にあたる基本オペレータの系列を一つにまとめたものである．よって，マクロオペレータの適用により，探索なしに複数の基本オペレータの適用を一度に行えるため，効率が向上することが期待される．

人工知能における最初のマクロオペレータは，STRIPS[Fikes 71]の学習バージョン[Fikes 72]における MACROPS であろう．STRIPS では，一度立てられたプランは，三角表(triangle table)と呼ばれる表現で記録され，オペレータの引数が変数化されることにより一般化が行われる．そして，次に過去のプランやその部分プランが適用できる問題が与えられたとき，過去のプランを用いて効率良い問題解決が可能になる．問題に適用できる過去の部分プランは，三角表から簡単に抽出される．ただし，原則的に STRIPS は，過去のプランをすべて残しておくため，後述する効用問題が生じるが，当時はそれが認識されていなかったようである．

通常，マクロオペレータは，一度問題を解いた解法例から，その部分系列(あるいは全体)を一般化することにより獲得される．この一般化の手法として，EBL が用いられる．ここでは，一般化手法よりも，どの部分系列を抽出するかということが問題となる．この問題は一般には解決できず，これまでいくつかのヒューリスティクスや実験的に評価を行って探索する方法が研究されている．

MACROPS 以降のマクロオペレータ学習において提案された手法を以下にあげる．

（1） **Korf のマクロオペレータ**

Korf は，8パズル，ルービックキューブなどでマクロオペレータを用いて探索なしに効率的に問題解決を行う手法を示した[Korf 85]．Korf は，マクロオペレータの狭義の定義として,「すでに達成されている副目標を覆すことなく，さらに一つの副目標を達成するオペレータ系列」という定義を用いた．もし，与えられた問題において，任意の副目標についてこのようなマクロオペレータが見つかるなら，副目標を順々に達成していけば，探索なしに問題が解ける．このようなマクロオペレータが存在するための条件が operator decomposability[Korf 85]であり，実際に8パズルなどではこの条件が満たされており，既存の探索手法によりこのようなマクロの自動生成が可能である．しかし，operator decomposability がどれほど広い領域で成り立つかは明確ではない．

（2） **Minton の T-Macro & S-Macro**

この研究は，おそらく最初の選択的マクロオペレータ学習であり,効用問題を発見した最初の研究である．Minton は，多数あるマクロオペレータの候補から選択的に学習することの重要性を示し，次の2種類のマクロを提案した[Minton 85]．

- **S-macro**：過去の解法履歴の共通部分系列から作られるマクロ．
- **T-macro**：解法例中で目標から遠ざかる部分系列をまとめたマクロ．

S-macro については，個数の上限が決められており，それを超えると古いマクロから削除される．また，T-macro により，水平線効果(horizontal effect)が避けられる．実験的に上記のマクロの選択的学習の有効性が確かめられた．

（3） **完全因果性による選択**

完全因果性「山田 89]とは，解法例からのマクロ抽出のためのヒューリスティクスで，次のように定義される．今，解法例であるオペレータ系列を $[O_1, \cdots, O_n]$ とする(O_i はオペレータ)と，その部分系列 S の任意の要素 O_i が，下の条件を満たすとき S は完全因果性を満たすという．

(条件) O_i は初期状態では適用不可能で，かつ O_1～O_{i-1} の適用後の状態では適用可能である．

完全因果性を満たす部分系列のみを一つのマクロオペレータにすることで，マクロオペレータの選択的学習が実現される．さらに，1次方程式，不等式，ロボットの行動計画の分野で，完全因果性により有効かつ少数のマクロのみが学習されることが実験的に示された[山田 89]．

ほかに，ヒューリスティクスを中心にした Iba の研究[Iba 89]，SOAR におけるチャンキングの研究がある．なお，SOAR については，本学会誌の[Soar 94]が詳しい．

なお,解法例からのマクロオペレータ選択の問題は，後述する効用問題と深く関わりあう．マクロオペレータ学習の研究も初期の一般化の研究から，どのように有効なマクロのみを選択するかという問題が最近の研究の中心テーマとなっている．

2・4 学習の実験的評価

効率化学習は，図1のように，それ単体で問題解決可能なオブジェクト知識をすでに持っているので，そのオブジェクト知識だけによる問題解決と，学習されたメタ知識を用いた問題解決を比較し，効率が向上するか否かを実験的に調べることにより，明確に学習の効果を評価できる．具体的には，同じ問題系列をオブジェクト知識だけの学習なしシステムとメタ知識を獲得する学習システムに与え，それらの問題をすべて解くのに要する CPU 時間などで比較を行う．このとき，これまでの研究では，まず学習システムに訓練例を与えて学習を行わせる訓練フェーズを実行し，その学習後に，学習なしシステムと学習システムに同じ問題を与えて比較を行う試験フェーズを行うというように，オフライン学習として評価している．つまり，そこでは，学習コストは評価されていない．しかし，実時間システムなどにおける効率化学習を考えると，オンライン学習を行わねばならず，学習コストこみでの比較が重要である．残念ながら，効率化学習におけるオンライン学習の研究はまだ少ない[山田 94]．

なお，以上のような比較において通常学習なしシステムの効率が非常に悪いため，問題1問当りに使える CPU 時間の上限を設けて評価を行う場合が多いが,この上限の取り方によっては，学習効果が逆転してしまうという報告がされている[Segre 91]．また，この問題に対し，仮説検定により，その効果を統計的に検証する研究がある[Etzioni 94]．

3．効 用 問 題

効率化学習を実験的に評価するとき，次のようなアイロニカルな現象が起こる．それは，メタ知識の学習を過剰に行った場合，そのメタ知識の適用を調べるコストが大きくなり，かえって学習なしシステムよりも効率が低下してしまうことである．この問題が「効用問題(utility problem)」と呼ばれ[Minton 88]，最近

の効率化学習研究の最重要テーマの一つである．オブジェクト知識だけで簡単に解ける問題が，悪質なメタ知識により計算コストが指数オーダになってしまうことが，STRIPSにおけるロボットの部屋の移動などの単純な領域でも簡単に起こり得る[Minton 88]ため，効用問題の解決は重要である．また，効用問題を帰納学習まで一般化する解釈もある[Holder 90]．次に，効用問題の解決法について紹介する．

3・1 初期の解決法

2•3節でのMintonの研究[Minton 85]では，学習されるマクロ数の上限を設定し，それ以上は学習しないことにより，効用問題に対処している．しかし，これはマクロの効用をまったく見積もらないやみくもな方法である(でも，結構良い結果が出たりもする)．

また，前述のPRODIGYでは，学習時に，その制御規則により省かれる節約時間(saving time)を一つの例から1回だけ測る．そして，その制御規則が適用されるごとに，その1回だけの計測値が割り当てられ，規則の適用ごとに新たに測られるマッチングコストの累積が，節約時間の累積よりも大きくなると，その制御規則は取り除かれる．この方法は，最初の1回の見積りの誤差が響き，有害なマクロを取り除くことができない場合がある[Minton 88]．

最近の効用問題の研究により，学習システム外の環境の構造として，訓練例の分布(問題分布)が着目され始め，興味深い知見が得られている．以降では，その問題分布と最近の効用問題の具体的な解決法について述べる．

3・2 問題分布を考慮した解決法

効率化学習の研究の進展とともに，学習された知識の効用は，どのような問題がどのような頻度で与えられるかという問題分布(problem distribution)に依存することがわかってきた．問題の与えられ方の確率分布は，学習オートマトンやPAC学習[Valiant 84]の分野ではすでに扱われてきたが，帰納学習やEBLに代表される経験的学習(empirical learning)の分野でほとんど扱われてこなかった．問題が学習システムにとっての環境からの情報であるとすると，問題分布はある意味で環境の構造を表しており，学習の効果が環境の構造に依存するというもっともらしい知見が確認される．このような問題分布を考慮した効用問題へのアプローチを紹介する．全体に，問題分布を取り入れて，山登り法で局所最適なメタ知識の集合を探索するものが多い．

GratchのCOMPOSER[Gratch 92]とGreinerのPALO[Greiner 92]は，ともに効用を評価関数として，山登り法で局所的かつ確率的に最適な制御規則集合を探索する．これらのシステムでは，効用が問題分布を考慮して計算される．この二つの手法の問題点として，マクロ集合の探索コストがある．特にPALOは非常に探索コストがかかることが報告されている．

Lairdは，並び換え(reordering)と展開(unfolding)により，局所最適な変換を山登り法で探索する方法を示し，そして問題分布を獲得する手法としてTDAG (Transition Directed Acyclic Graph) を提案した[Laird 92]．

また，筆者も問題系列の一部についてマクロオペレータ学習を行い，そこから得られた情報のみから問題分布の推定，そして局所最適なマクロ列を決定する方法を示した[山田 94]．この手法は，山登り法を用いるのではなく，学習されたマクロ数の線形オーダに制限された探索空間内の最適マクロ集合を探索する．よって，学習は十分高速であり，実時間システムなどにおけるオンライン学習の評価にも耐え得るという特徴を持つ．

4．スケジューリングへの応用例

残念ながら効率化学習を，積木の世界や簡単なプラニングなどの実験の域を超えた領域に応用した例は，筆者の知る範囲ではあまり見当たらない．ここでは数少ない応用例のうちNASAでのスケジューリングへの応用[Zweben 92]を紹介する．

対象領域は，制約に基づくスケジューリングであり，具体的にはスペースシャトルのペイロード処理(テスト，検査，カーゴへの荷積みなどのタスク)とグラウンド処理(シャトル発射のための準備)のスケジューリングの二つである．それぞれの対象に対し，問題の性質から，オブジェクトレベルの問題解決として，forward checkingつきのバックトラックと，制約違反を繰り返し修正していくiterative repair法の二つの解法が使われた．

学習方式は，ある説明でいくつの訓練例が説明できるかでその説明の信頼度を評価し，その評価がしきい値以上になった場合にのみ学習するという選択的学習であるPEBL(Plausible EBL)を用いている．バックトラックで解くペイロード処理では，時制の制約と資源の制約のどちらから具体化するかという変数順序(varible ordering)が，iterative repair法で解くグラウンド処理では，多情報でコスト高の修正ヒューリス

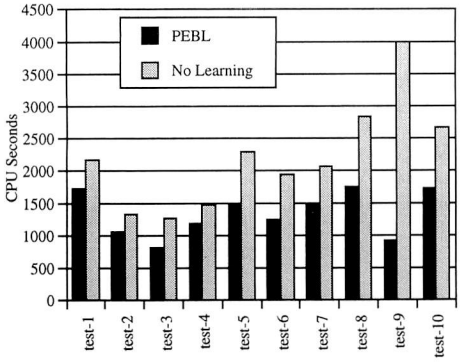

図4　ペイロード処理の結果[Zweben 92]

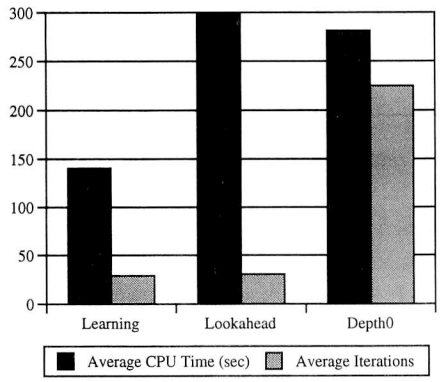

図5　グラウンド処理の結果[Zweben 92]

ティクス Lookahead と少情報でコスト安のヒューリスティクス depth0 いずれを選択するかが，EBL により学習される．

それぞれの学習後のシステムと学習なしのシステムに同じ問題を与えたときのパフォーマンスの比較結果を図4と図5に示す．図4のペイロード処理では，19〜77%の範囲(平均34%)で効率化が実現されている．なお，図中には示していないが，選択的でない学習だと，すべてのテストにおいて，学習なしシステムよりも効率が低下している．

次に，図5のグラウンド処理での結果を見る．横軸の Lookahead と depth0 が，選択なしにそれぞれのヒューリスティクスのみを用いた場合で，縦軸はある誤差範囲への収束にかかったコスト(CPU 時間と iteration 回数)である．CPU 時間で，ほぼ50%の高速化が図られていることがわかる．

以上は，一つの領域における結果であるが，他の実験[Minton 88]を見ても，効率化学習では，速度が2〜数倍のオーダで高速化される程度であり，1桁さらには2桁早くなるような例はないように思える．もちろん，訓練および試験フェーズで与える問題分布に依存するが，よほど問題分布が偏っていない限り，1桁以上の効率化は困難であろう．

5. ま　と　め

本稿では，効率化学習の方法論から応用例までを概観した．効率化学習(あるいは EBL)は，帰納学習と比べると，学習された知識が領域理論から明示的に理解できるために，学習された知識の正当性が保証されている反面，帰納的飛躍(inductive leap)などのあやしげな手続きを欠き，いわゆる学習としてのおもしろ味に欠けるという見方もある．しかし，効用問題の発見により，学習された知識の正当性からは，学習システムとしての効率化が保証されないこと，さらに効率化は問題分布(環境の構造)と学習システムの関係から決まるという興味深い知見が得られており，この方向からのアプローチが，今後この分野をおもしろくすると信じる．

◇参　考　文　献◇

[DeJong 86] DeJong, G. and Mooney, R.: Explanation-Based Learning: An Alternative View, *Machine Learning*, Vol. 1, No. 2, pp. 145-176 (1986).

[Etzioni 94] Etzioni, O. and Etzioni, R.: Statistical Methods for Analyzing Speedup Learning Experiments, *Machine Learning*, Vol. 14, No. 3, pp. 333-347 (1994).

[Fikes 71] Fikes, R. E. and Nilsson, N. J.: STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Artif. Intell.*, Vol. 2, No. 3/4, pp. 189-208 (1971).

[Fikes 72] Fikes, R. E., Hart, P. E. and Nilsson, N. J.: STRIPS: Learning and Executing Generalized Robot Plans, *Artif. Intell.*, Vol. 3, No. 4, pp. 251-288 (1972).

[Gratch 92] Gratch, J. and DeJong, G.: COMPOSER: A Probabilistic Solution to the Utility Problem in Speed-up Learning, *AAAI-92*, pp. 235-240 (1992).

[Greiner 92] Greiner, R. and Jurisica, I.: A Statistical Approach to Solving the EBL Utility Problem, *AAAI-92*, pp. 241-248 (1992).

[Holder 90] Holder, L. B.: The General Utility Problem in Machine Learning, *ML-90*, pp. 402-410 (1990).

[Iba 89] Iba, H.: A Heuristic Approach to the Discovery of Macro-operators, *Machine Learning*, Vol. 3, No. 4, pp. 285-318 (1989).

[Korf 85] Korf, R. E.: A Weak Method for Learning, *Artif. Intell.*, Vol. 26, No. 1, pp. 35-77 (1985).

[Laird 92] Laird, P.: Dynamic Optimization, *ML-92*, pp. 263-272 (1992).

[Michalski 83] Michalski, R. S.: A Theory and Methodology of Inductive Learning, Michalski, R. S., Carbonell, J. G. and Mitchell, T. M. (eds.), *Machine Learning—An Artificial Intelligence Approach—*, pp. 83-134, Tioga (1983).
電総研人工知能研究グループほか訳：「帰納学習の理論と方法論」，知識獲得入門—帰納学習と応用—，共立出版 (1987).
[Minton 85] Minton, S.: Selectively Generalizing Plans for Problem Solving, *IJCAI-85,* pp. 596-599 (1985).
[Minton 87] Minton, S. and Carbonell, G.: Strategies for Learning Search Control Rules: An Explanation-based Approach, *IJCAI-87,* pp. 228-235 (1987).
[Minton 88] Minton, S.: *Learning Search Control Knowledge—An Explanation-Based Approach—*, Kluwer Academic Publishers, Boston (1988).
[Mitchell 86] Mitchell, T. M., Keller, R. M. and Kedar-Cabelli, R. T.: Explanation-Based Generalization: A Unifying View, *Machine Learning,* Vol. 1, No. 1, pp. 47-80 (1986).
[沼尾 88] 沼尾正行：説明に基づく学習—領域固有の知識を用いたアプローチ—，人工知能学会誌，Vol. 3, No. 6, pp. 704-711 (1988).
[Segre 91] Segre, A., *et al.*: A critital look at experimental evaluations of EBL, *Machine Learning,* Vol. 6, No. 2, pp. 183-196 (1991).
[Soar 94] 小特集「Soar プロジェクト」，人工知能学会誌，Vol. 9, No. 4, pp. 478-504 (1994).
[Valiant 84] Valiant, L. G.: A Theory of the Learnable, *Commun. ACM,* Vol. 27, No. 11, pp. 1134-1142 (1984).
[山田 89] 山田誠二，辻　三郎：完全因果性によるマクロオペレータの選択的学習，人工知能学会誌，Vol. 4, No. 3, pp. 321-329 (1989).
[山田 91] 山田誠二：EBL にとって説明とは何か，認知科学の発展，Vol. 4 (1991).
[山田 94] 山田誠二：EBL 効用問題の実用的解決とそのオンライン評価，人工知能学会誌，Vol. 9, No. 3, pp. 393-399 (1994).
[Zweben 92] Zweben, *et al.*: Learning to improve constraint-based scheduling, *Artif. Intell.*, Vol. 58, pp. 271-296 (1992).

著 者 紹 介

山田　誠二(正会員)

1984 年大阪大学基礎工学部卒業．1989 年同大学院博士課程修了．同年，基礎工学部システム工学科助手．1991 年より大阪大学産業科学研究所講師，現在に至る．工学博士．人工知能，特に，効率化学習，即応プラニング，ロボットナビゲーション，マルチエージェント系に興味を持つ．情報処理学会，日本認知科学会，日本ソフトウェア科学会，AAAI，IEEE 各会員．