

記憶に基づく学習への帰納学習の埋め込みについて

畝見 達夫

長岡技術科学大学 工学部 計画・経営系

unemi@voscc.nagaokaut.ac.jp

粒度の小さな離散時系列データを対象とする「記憶に基づく学習」のメカニズムに帰納学習の手法を埋め込むための1つの枠組を提案する。対象となる問題は実数ベクトルを入力としシンボルを出力とする補強入力の遅れを伴う補強学習問題である。具体的には、我々が先に提案した「記憶に基づく時系列学習法」に埋め込むための、記憶されたデータに対する要素レベルの一般化と、時系列の分節化を提案する。これらの機構の組合せは低レベル記憶からのルール生成につながるものである。環境適応問題を用いた計算機シミュレーションにより帰納学習導入に伴う性能に变化が観察された。

On Embedding an Inductive Learning Schema into a Memory-Based Learning System

Tatsuo UNEMI

Department of Planning and Management Science, Nagaoka University of Technology

1603-1 Kamitomioka-cho, Nagaoka, Niigata 940-21, Japan

unemi@voscc.nagaokaut.ac.jp

This paper proposes a framework to embed an inductive learning mechanism into a memory-based learning system for discrete time-sequence of small grain size. The learning task follows a reinforcement learning scenario, where the input is a vector of real numbers, the output is a symbol and the reinforcement is delayed. Concretely, we presents two methods, which are to generalize the memorized data elements, and to cluster the memorized sequences. Combination of these methods will lead to symbolic rule generation. The result of computer simulation shows the difference of performance caused by introducing inductive learning schema.

1 はじめに

計算機の処理能力の向上に伴い、学習研究のテーマにも変化が起きつつある。その1つが「記憶に基づく[1,2]」あるいは「実例に基づく[3]」接近法、すなわち、多くの訓練例を加工せずに記憶し、問題解決の時点で、なんらかの類似尺度に従った類似知識検索を行なうことにより、未知の状況に対応しようとするものである。Samuel 先駆的な研究[4]以来、綴りから発音への変換[1]、自然言語の翻訳[2]、診断システム[5]、簡単なロボット制御[6]、ロボットマニピュレータの制御知識の獲得[7]などの応用例が報告されている。これらの手法は、ある程度の大きさの記憶容量と高速な計算能力を必要とする一方、計算の複雑性を多項式オーダーで押えられ、コネクションマシンなどの大規模並列計算機上での実行に適するという点においても注目に値する[5]。訓練例を一般化することによって規則あるいはパターン記述を生成するといった帰納推論[8]あるいは帰納学習[9]の方法とは様々な意味で対照的である[10]。

我々は先に、主に生体の環境適応行動をシミュレートするモデルを構築する立場から、記憶に基づく学習の枠組に基づいた無限長離散時系列を対象とする学習機構を提案した[11]。ここでは、その学習機構を拡張する形で、記憶に基づく学習への帰納学習の埋め込みの1つの枠組について提案する。先に提案した学習機構では個々の記憶要素の粒度が小さいため、記号処理機構への発展やエキスパートシステムへの応用を考えた時、何らかの抽象化機能の導入が必要となる[12]。ここではその第1歩として、記憶されたデータに対する要素レベルの一般化と時系列の分節化の2つの枠組を提案する。これらの機構の組合せは低レベル記憶からのルール生成につながるものである。

以下、問題の定式化、基礎となる記憶に基づく学習機構の概説、帰納学習の埋め込み、計算機シミュレーションによる学習性能を示す。

2 無限長時系列学習問題

先の研究で想定された学習主体は、餌を求めて2次元平面の仮想世界を歩き回る Beer の人工昆虫[13]のような「虫」であった。

入力とは虫の視覚に対応した10次元のベクトルである。各要素は視野に入った物体の種類と距離を反映する-9から9の整数である。出力は虫の方向転換に対応し、-3から3の整数を値とする。学習の目的は世界に配置されたゴミに触らずに、餌を食べることである。物体に触れた時にそれがゴミか餌かの情報が得られる。

学習問題は、入出力と目的の構造によって決まるが、ここで扱う問題をその意味から定式化すればつぎようになる。入力は2から数十次元の数値ベクトルを1つの時点の要素とする無限長の離散時系列、出力は数個の要素からなる離散値の1つを1時点の要素とする同様の離散時系列である。目的は3個の要素のうちの1つを値とする特別の入力によって与えられる。学習システムの性能は、この目的入力の列の中に特定の値が占める密度によって与えられる。形式的に記述すればつぎようになる。

$$\begin{aligned} \text{入力 } X_t &= (x_{t1}, x_{t2}, \dots, x_{tn}) & (x_i \text{ は実数}) \\ \text{出力 } Y_t &\in \{-3, -2, -1, 0, 1, 2, 3\} \\ \text{目的入力 } R_t &\in \{\text{"ゴミ"}, 0, \text{"餌"}\} \end{aligned}$$

このような問題設定はニューラルネットに基づくアプローチによって研究が進められている補強学習法 (reinforcement learning method)[14,15,16]において対象とされてきた問題の一種と考えられる。

3 記憶に基づく離散時系列学習の概要

まず、本論文で提案する枠組のもととなる「記憶に基づく学習」について概説する。詳しくは先の報告[11]を参照されたい。

3.1 記憶の基本構造

記憶は記憶要素と名付けられたノードを時間方向に連結した列の集合である。各記憶要素は1ステップにおける入出力データを記憶するために用いられる。各記憶要素には、付加的な情報として重み、価値などが属性として記録される。

記憶要素と重みの対によって構成される想起表と呼ばれる有限集合を用いる。この集合は最大10個程度の要素からなる。各要素の重みは、記憶されたデータと現状との間の、ある意味での類似性を表すものであり、想起表自身は各ステップにおける虫の現状認識を表現するものである。

想起表要素の1つを計画と呼び、その運動記憶データが次のステップにおける出力データの候補として扱われる。計画は存在しない場合もある。想起表と計画の管理方法については後に述べる。

3.2 記録と忘却

各ステップにおいて入出力データを書き込んだ記憶要素を記憶に追加する。この時の初期重みは 0.5 とする。実際には記憶量の爆発を防ぐため、記憶要素の最大数をあらかじめ制限しておく。つまり、その最大数以上の実行ステップ数を経た後では、新たなデータを記憶するためには、既存の記憶データの1つを忘却しなければならない。忘却する記憶の選択は重みによって決定する。すなわち、最も重みの小さな記憶が忘却の候補となる。すべての記憶要素の重みは各ステップにおいて $(0, 1)$ の定数 η を掛けることで減少させる。後で述べる実験では $\eta = 0.98$ とした。後の 3.4 節で述べるように、反省の段階でも重みに対する別種の修正が行なわれる。

実行ステップ数が記憶容量と同じ数に到達するまでは、記憶は1本の連鎖であるが、それを越えると忘却によって結合が分断されるため複数の連鎖の集合となる。分断のために不要となった記憶要素、たとえば他の要素との結合をまったくもたないものなどは同時に忘却する。

3.3 類似経験の想起

本学習機構における重要な技法の1つは、記憶中からの類似データの想起による現状把握である。本学習機構において実際に用いる、データ x と y と間の類似度 $S(x, y)$ は次のように定義される。

$$S(x, y) = 1 - \frac{1}{2n} \sum_{i=1}^n \frac{(x_i - y_i)^2}{V_{x_i}} \quad (1)$$

ここで n は次元の数すなわちベクトルの要素数、 x_i および y_i はそれぞれ x および y の i 番目の要素、 V_{x_i} は i 番目の要素についての記憶中の全データに関する分散の近似値である。分散の近似値はデータの2乗の平均と平均の2乗の差として算出できるので、各ステップでデータとその2乗の平均の近似値を保持することにより容易に算出できる。データ値の近似値はつぎのような各ステップでの加重平均により計算する。

$$E_t = \begin{cases} \frac{1}{t} E_{t-1}(t-1) + x_t & \text{if } t < k \\ \frac{1}{k} E_{t-1}(k-1) + x_t & \text{otherwise} \end{cases} \quad (2)$$

ここで、 k はある程度大きな整数であり、 k ステップ目までは正確な平均値をとり、それ以降は近似値となる。実験では $k = 50$ とした。

類似度は最大値が1となり、平均値はデータの分布に関係なく次式に示すとおりほぼ0となる。

$$E_{St} = 1 - \frac{1}{2n} \cdot \frac{2}{t(t-1)} \cdot \sum_{i=2}^N \sum_{j=1}^{i-1} \sum_{k=1}^n \frac{(x_{ik} - x_{jk})^2}{V_{x_k}} = 1 - \frac{1}{2n} \cdot \frac{2tn}{t-1} \approx 0 \quad (3)$$

想起表要素の初期重みは、現ステップの入力データベクトル c と想起されたデータベクトル m の間の類似度 $S(c, m)$ とする。各ステップでの現状を把握するため、想起表要素に含まれるすべての記憶要素を、その記憶要素とリンクで結合された次ステップの記憶要素で置き換える。このとき想起表要素の重みを、置き換わった新たなデータと新たに入力されたデータとの間の類似度との次式のような加重和を取ることによって変更する。

$$RTE_i W_t = \alpha \cdot RTE_i W_{t-1} + (1 - \alpha) \cdot S(c, m_i) \quad (4)$$

ここで $RTE_i W_t$ は時刻 t における i 番目の想起表要素の重み、 m_i は連想された記憶要素の入力データ、 α は何ステップ分を文脈として考慮するかを示す $(0, 1)$ の係数である。 α の値が大きくなるほど長い文脈を優先して考慮することになる。もし値が0なら文脈は考慮されない。実験では α の値は 0.4 とした。

加えて、記憶中から高い類似度を持つデータを検索し、次ステップにおける想起表要素の候補とする。既存の想起表要素と新たな検索データを合わせた集合の中から重みの大きい方から10個以内を次ステップの想起表要素とする。

3.4 予測と反省

各記憶要素に期待される価値 (EPV) を割り当て、その値から将来に得られそうな価値の感触を得ることにより、予測を行なう。値の割り当ては、虫が餌あるいはゴミに遭遇した時点で、直前に記憶されたノードからリンクを逆方向にたどりながら、同時に割り当て値を徐々に減じながら、既に EPV が割り当てられたノードに到達するか、または割り当て値がある閾値を下回るまで行なわれる。記憶要素 ME_{c-} の新たな EPV は形式的にはつぎのように表現できる。

$$ME_{c-}EPV = \beta^i \quad (5)$$

ここで c は現在のステップ番号、 β は $(0, 1)$ の実数定数である。実験では $\beta = 0.85$ 、閾値を 0.05 とした。

また、計画に従って行動した結果、ゴミに遭遇した場合、計画に連なる記憶要素 ME_i の EPV はつぎのように修正される。

$$ME_iEPV_i = ME_iEPV_{i-1} - (1 - \gamma_1^{RTE_pW}) \cdot \gamma_2^s \cdot (ME_iEPV_{i-1} + 1) \quad (6)$$

ここで RTE_pW は現在の計画となっている想起表要素の重み、 s は計画の記憶要素までのリンクの数、 γ_1 および γ_2 は $(0, 1)$ の実数定数である。実験では $\gamma_1 = 0.8$ 、 $\gamma_2 = 0.85$ とした。上式は記憶要素が現在の計画に近く、現在の状況に類似していればいほど、EPV の値を大きく減少させることを示している。同時に記憶要素の重みをつぎの式に従って変更する。

$$ME_iW_t = \begin{cases} ME_iW_{t-1} + \rho_1 \cdot \rho_2^s \cdot (1 - ME_iW_{t-1}) & \text{餌の場合} \\ ME_iW_{t-1} - (1 - \rho_1) \cdot \rho_2^s \cdot ME_iW_{t-1} & \text{ゴミの場合} \end{cases} \quad (7)$$

ここで ρ_1 および ρ_2 は $(0, 1)$ の実数定数である。実験では $\rho_1 = 0.5$ 、 $\rho_2 = 0.95$ とした。重みの値を絶えず 0 から 1 の間に保持しながら、餌に遭遇した場合は重みを増し、ゴミの場合には減らす。このように、より信頼のおけるデータ系列は重みを大きくし、信頼性の低い系列は重みを小さくすることによって、有用な系列が記憶中に残るようになる。

3.5 意思決定

虫はステップ毎に自身の行為を決定する。もし、計画が存在すれば決定はその内容に依存するが、計画が無い場合は一時的な価値の予測によって決定する。ただし、もし単一の行為をある程度の根拠をもって選択することができなかったならば、有限の可能な行為の候補、つまり -3 から 3 までの整数の中からランダムに選ぶことになる。

想起表中の記憶要素をリンクを 1 つたどることによって次の記憶要素に置き換え、計画が存在する場合は計画に含まれる記憶要素の出力データを選択する。計画の採用と撤回は、想起表要素の重み RTE_iW と、想起表要素中の記憶要素の期待される価値 $RTE_iME.EPV$ を参照し、つぎのような式によって計算される値を基に決定する。

$$m_i = \min(RTE_iW, RTE_iME.EPV) \quad (8)$$

現在の計画に関する m_i の値がある閾値 θ を下回った場合には計画を撤回する。 θ は虫の空腹度を表わす $(0, 1)$ の実数であり、餌が得られるまで各ステップ毎に増加し、餌が得られた時点で小さな値に初期化される。実験では θ の初期値を 0.05 とし、各ステップ毎に値が 1 を越えるまで 1.1 を掛ける。計画が撤回された場合、想起表要素の中から 1 つを新たな計画として採用すべく各想起表要素の m_i の値を調査する。 θ より大きな値が見つからなければ計画は空となる。想起表要素の中で最大の m_i をもつものがいつも計画になるのではないという点に注意されたい。計画として一旦採用された要素は、その m_i の値が θ を下回るかあるいは想起表から消去されるまでは計画であり続ける。

計画が存在しない場合には、計画が撤回された場合と同様に、どの想起表要素が計画として最良かを調査する。計画の採用に失敗した場合には、すべての想起表要素について次のステップの記憶要素の EPV を用いて m_i の値を計算し同一の運動命令についての m_i の総和を取る。この中で相対的に大きな値をもつ命令を出力データとして選ぶ。ただし、その値が θ より小さければランダムに選ぶこととする。初期状態では計画は存在せず、想起表も空なので出力データはランダムに選ばれる。

4 帰納学習の埋め込み

帰納学習の基本操作は正の例からの一般化と負の例からの特殊化である[9]。本研究で対象とされる問題では、計画が成功した時、すなわち、予測通りに正の価値をもつ目的入力力が得られた場合、それまでの計画は正の例となり、計画が失敗した時、すなわち、予測に反して負の価値を持つ目的入力力が得られた場合、負の例となる。正の価値を期待したにもかかわらず、目的入力力が 0 であった場合も、失敗とみなすことができる。

本研究で扱われるデータは、多次元実数ベクトルの無限長時系列である。本来は、データの単位が不明確なため、帰納学習の手法を直接用いることが不可能であったが、計画を1つの単位とみなすことによって、複数の例からの帰納が可能となった。1つの計画は、あるステップ数に渡る系列である。これは、時間方向と空間方向の2次元パターンとみなすこともできる。ここでは、個々の要素を区間にするこゝによる一般化と、無限長系列の分節化を提案する。

4.1 要素データの一般化

予測通りに正の価値をもつ目的入力を得られた場合、計画を経験との比較に基づいて一般化する。具体的には、ステップ毎に対応する個々の記憶要素中のデータを両方の値を含むような最小の区間に変更する。もし、計画側のデータが既に区間となっており、かつ経験側のデータがその区間に含まれない場合は、経験側のデータが含まれるような最小限の拡張を行なう。現在の計画に含まれる記憶要素だけでなく、そこから逆向きにリンクを辿り、計画として採用された最後の記憶要素に至るまでのすべての要素を一般化の対象とする。

区間化されたデータに対する類似度は、つぎのような式によって計算する。

$$S(x, y) = 1 - \frac{1}{2n} \sum_{i=1}^n (D(x_{i\min}, x_{i\max}, y_i) + B(x_{i\max} - x_{i\min}, 2\sqrt{3V_{x_i}})) / V_{x_i} \quad (9)$$

$$D(x_{\min}, x_{\max}, y) = \begin{cases} (x_{\min} - y)^2 & \text{if } y < x_{\min} \\ 0 & \text{if } x_{\min} \leq y \leq x_{\max} \\ (y - x_{\max})^2 & \text{if } x_{\max} < y \end{cases} \quad (10)$$

$$B(w, p) = \frac{w^3}{6p} - \frac{w^2}{2} + \frac{wp}{2} \quad (11)$$

ここで、 $x_{i\min}$ および $x_{i\max}$ は、 x の i 番目の要素の最小値と最大値である。すなわち、区間に含まれる場合は一定の類似度とみなし、含まれない場合は、 n 次元ユークリッド空間上において、区間を表す矩形領域の境界面までの距離によって値を決める。 $B(w, p)$ は、 $x_{i\min}$, $x_{i\max}$, y が、それぞれ区間 $[0, p-w]$, $[w, p]$, $[0, p]$ の一様分布に従うときに、類似度の平均が0となるようにするためのバイアスである。(11)式の導出については付録を参照されたい。区間 $[x_{i\min}, x_{i\max}]$ が大きくなると、多くのデータに対して比較的高い類似度を示すようになる反面、類似度の最大値は低く押えられる。このような性質から考えると、区間化によって一般化されるというよりも、「あいまいにされる」といったほうが適切かもしれない。

4.2 系列の分節化

各記憶要素に、類似検索および計画としての採用の可否を決める指標を属性として割り当て、その値を成功、失敗に応じて増減することにより、系列にある意味での切れ目を付ける。各記憶要素に関する類似検索の指標 $ME_i.SR$ および計画採用の指標 $ME_i.PA$ を、つぎの式に基づいて逐次的に変更する。

$$ME_i.SR_i = \mathcal{M}(ME_i.SR_{i-1}, \langle \text{検索時の類似度} \rangle) \quad (12)$$

$$ME_i.PA_i = \mathcal{M}(ME_i.PA_{i-1}, \langle \text{計画採用時の重み} \rangle) \quad (13)$$

$$\mathcal{M}(x, a) = \begin{cases} a + x - ax & \text{if 餌に遭遇した.} \\ (1-a) \cdot x & \text{if ゴミに遭遇した.} \\ (1-a/2) \cdot x & \text{if * (以下の本文参照)} \\ x & \text{otherwise} \end{cases} \quad (14)$$

上式中の * は、計画が途切れたか、あるいは餌を期待したが何も得られなかった場合である。 $ME_i.SR$ の値が閾値を下回った記憶要素は、類似記憶検索の対象から除外され、 $ME_i.PA$ の値が閾値を下回った記憶要素は、計画採用の対象から除外される。閾値を下回るまでは、成功、失敗に応じてそれぞれの指標の値が増減されるが、一旦閾値を下回った後は、類似記憶検索あるいは計画採用の対象とはならないので、指標の値が変更されることはない。実験では閾値の値を0.2とした。

4.3 ルールとしての記憶系列

分節化された記憶系列は、条件部と実行部からなるルールとして見る事ができる。すなわち、類似記憶検索可能な要素から計画採用可能な要素の前までの系列が条件部、計画採用可能な要素から「餌」までの系列が実行部に相当す

る。つまり、類似記憶検索をルール条件部の照合の開始、計画採用を照合の成功と実行の開始に当てる。ただし、実行中にも状況と記憶との照合が行なわれるため、通常のプロダクションルールとは異なり、条件に合わなければ実行が中断されることになる。

ルールの生成過程を見ると、初期状態ではすべての記憶要素が類似記憶検索可能かつ計画採用可能であり、それらの可能性を示す指標が徐々に更新されるという点から言えば、系列のルール化は徐々に行なわれることになる。つまり、条件部、実行部が明確に分けられるという意味でのルールは生成されない可能性もある。

5 実験結果

先の報告[11]と同じ環境設定の下で、計算機シミュレーションによる学習性能の調査を行なった。規則的な配置とランダムな配置の2種類の環境について、1. 帰納学習を埋め込まない場合、2. 一般化を行なった場合、3. 系列の分節化を行なった場合、4. 一般化と系列の分節化を行なった場合の各々について500サイクルの実行を100回ずつ行なった。図1に典型的な虫の足跡を示す。

図2に学習曲線を示す。縦軸は、 j サイクル目までの「餌」に遭遇した数を N_j^+ 、「ゴミ」に遭遇した数を N_j^- としたときの $(N_j^+ - N_j^-)/j$ の値である。

規則的な配置については帰納学習を用いた方が立ち上がりが早くなっている。ランダムな配置では一般化の導入によって性能が落ちる傾向が見られる。これらの現象は、つぎのように説明できる。帰納学習によって規則性の発見が行なわれるため、規則的な配置では、帰納学習の効果が現れるが、ランダムな配置の環境では逆に過剰一般化が起こりやすくなるため発見した規則が誤った偏見として働き、性能を落す結果となる。

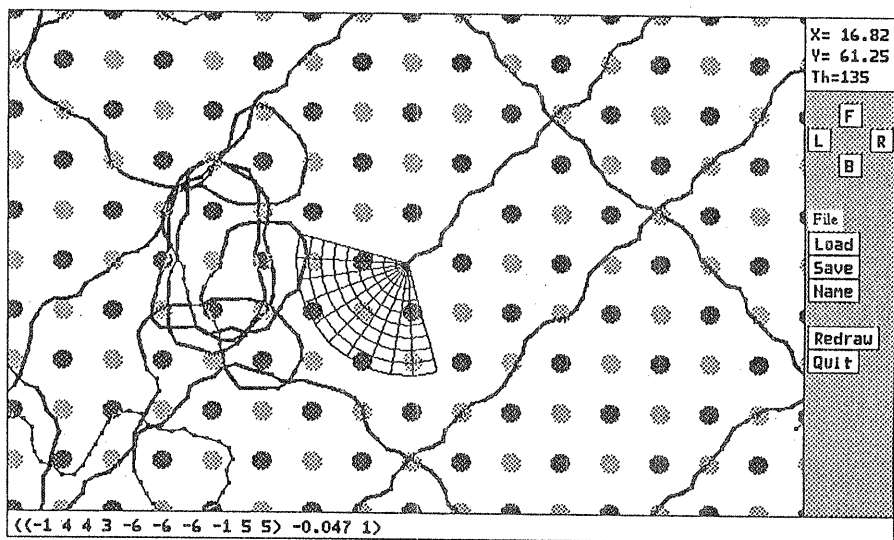
6 おわりに

本研究は様々な意味で遺伝アルゴリズム[17]と類比できる。成功や反省に基づく重みと価値の逆伝播は、パケツリレーアルゴリズムと類似している。パケツリレーアルゴリズムでは、ルールの連鎖によってタスクが成功した場合に同様の逆伝播メカニズムを使って信頼度の配分を行なっている。しかし、失敗に対しては逆伝播を用いずに、活性化した時点で信頼度を少し減少させることで同様の効果を実現している。これは、複数のルール系列が同時に活性化することを前提としているため、競争に負けたルール系列は、特に逆伝播を行なわなくとも結果的に信頼度が下がるよう仕組まれている。このようなシステムにおいて、負の信頼度の逆伝播を明示的に用いると、失敗したすべてのルール系列に対して逆伝播を開始する必要が生じるため、計算効率の上から、あるいは認知モデルとしての妥当性の上から考えて適切でない。しかし、本システムでは活性化は即実行となるため、一度に活性化する記憶は唯一であり、負の信頼度の明示的な逆伝播を妨げる理由はない。

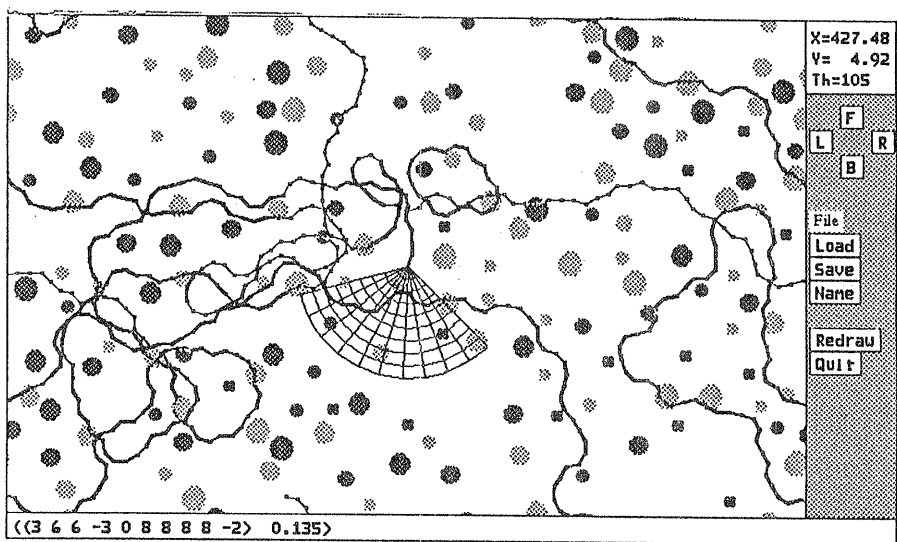
また、個々の記憶を、入力を条件部、出力を実行部とするルールとみれば、システム全体を分類子 (classifier) システムのように見ることもできる。おそらく、同様の問題に適用可能な分類子システムの枠組に基づいた学習システムを構築し、学習性能の比較を行なうことは有益であろう。

ここでは、要素の一般化と、系列の分節化という2つの方法について述べたが、ルール生成のためには、さらに別の、たとえば「まとめあげ」と「分解」[18]といった手法との統合も有効と考えられる。

本研究の主要な目的の1つは、何らかの意味でのデータの抽象化を導入することであった。記号主義からの批判として、この種の粒度の小さなデータを直接操作するアプローチでは様々な問題解決などにおける人間の知能を説明するのが困難であるという指摘が考えられるが、筆者は抽象化の導入によって人間知能への接近の糸口がつかめると考える。たとえば、言語の創出モデルへの発展が可能かもしれない。また、抽象化によって記号を創出し、ある種の説明を生成することも可能と思われる。説明の生成はエキスパートシステムへの応用に際しては重要な機能である。しかし、記号表現されたルールの生成は今後の課題である。



規則的な配置



ランダムな配置

図 1: 典型的な虫の足跡. (500 歩分. 図の中央が 500 歩目の位置である. 灰色の小円は「餌」, 黒の小円は「ゴミ」, 扇型は虫の視野を表す.)

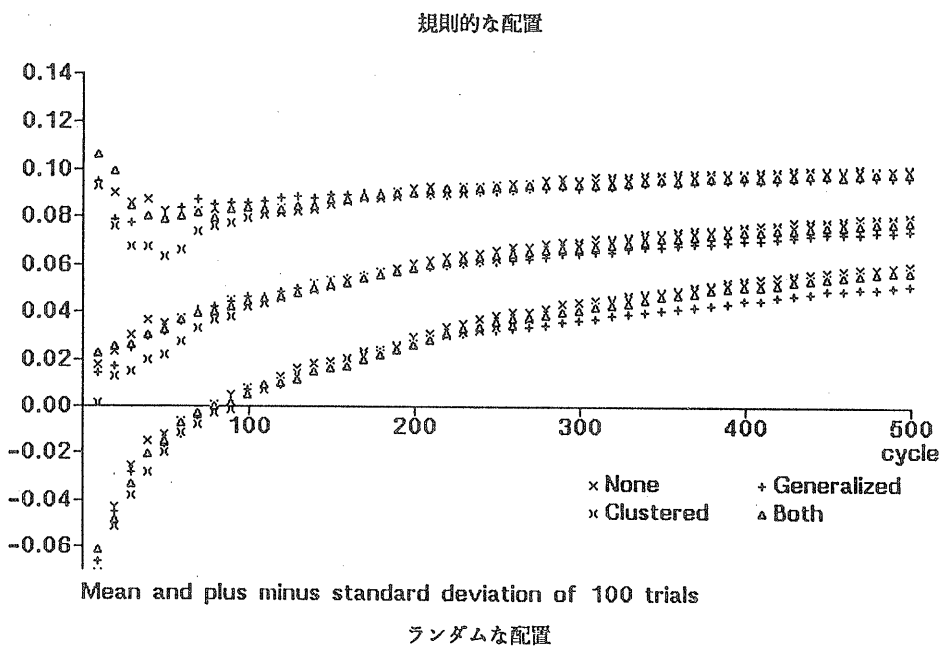
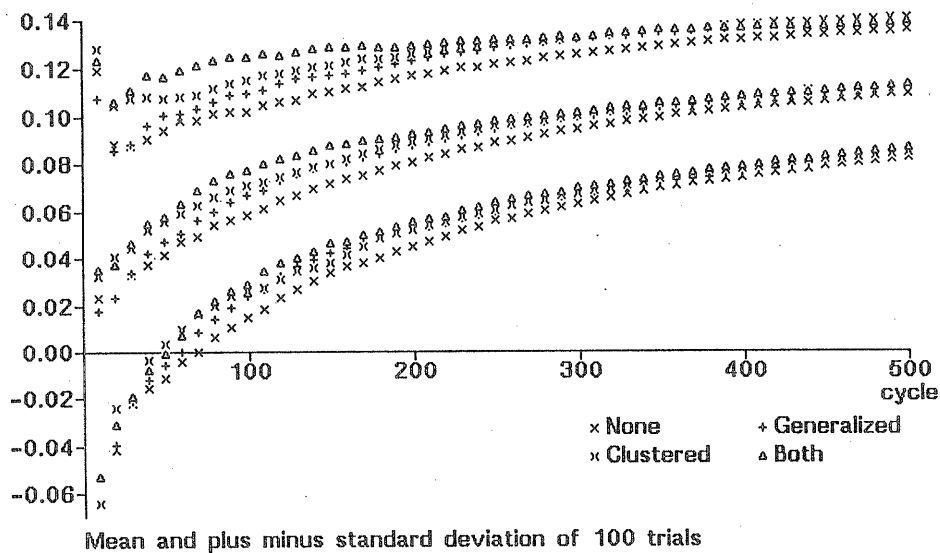


図 2: 学習曲線. (100 回の試行における平均値と, 平均値±標準偏差.)

参考文献

- [1] Stanfill, C. and D. Waltz: Toward Memory-Based Reasoning, *Communications of the ACM*, Vol. 29, pp. 1213-1228 (1986).
- [2] 佐藤理史: 記憶に基づく翻訳 II, 情報処理学会人工知能研究会報告 70-3 (1990).
- [3] Aha, D. W., D. Kibler and M. K. Albert: Instance-Based Learning Algorithm, *Machine Learning*, Vol. 6, pp. 37-66, (1991).
- [4] Samuel, A. L.: Some Studies in Machine Learning Using the Game of Checkers, *IBM Journal on Research and Development*, Vol. 3, pp. 210-229, (1959).
- [5] Waltz, D. L.: Applications of the Connection Machine, *IEEE Computer*, Vol. 20, pp. 85-97, (1987).
- [6] Mason, M. T., A. D. Christiansen and T. M. Mitchell: Experiments in Robot Learning, *Proceedings of the Sixth International Workshop on Machine Learning*, pp. 141-150, (1989).
- [7] Moore, A. W.: Acquisition of Dynamic Control Knowledge for a Robotic Manipulator, *Proceedings of the Seventh International Conference on Machine Learning*, pp. 244-252 (1990).
- [8] Angluin, D. and C. H. Smith: Inductive Inference: Theory and Methods, *Computing Surveys*, Vol. 15, pp. 237-269, (1983).
- [9] Michalski, R. S.: A Theory and Methodology of Inductive Learning, in Michalski, R. S., J. G. Carbonell and T. M. Mitchell, T. M. (Eds.), *Machine Learning: An artificial intelligence approach*, Palo Alto, CA: Tioga, (1983).
- [10] 佐藤理史: Memory-based アプローチと規則学習, 「学習のパラダイムとその応用」シンポジウム論文集, 情報処理学会, pp 69-77, (1989).
- [11] 畠見達夫: 記憶に基づく離散時系列の学習と環境適応シミュレーションへの応用, *Workshop on Learning 1991*, 北海道札幌市手稲, (1991).
- [12] 畠見達夫: 抽象化リンクによる信号と記号の融合, 日本認知科学会, 第5回大会発表論文集, pp 50-51, (1988).
- [13] Beer, R. D.: Intelligence as Adaptive Behavior: An Experiment in Computational Neuroethology, Academic Press (1990).
- [14] Barto, A. G., R. S. Sutton and C. W. Anderson: Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems, *IEEE Transactions on System, Man and Cybernetics*, Vol. 13, No. 5, pp. 834-846, (1983).
- [15] Kaelbling, L. P.: Learning Functions in k -DNF from Reinforcement, *Proceedings of the Seventh International Conference on Machine Learning*, pp. 162-169 (1990).
- [16] Whitehead, S. D. and D. H. Ballard: Active Perception and Reinforcement Learning, *Proceedings of the Seventh International Conference on Machine Learning*, pp. 179-188 (1990).
- [17] Holland, J. H., K. J. Holyoak, R. E. Nisbett and P. R. Thagard: Induction, MIT Press (1986).
- [18] 早勢欣和, 畠見達夫, 川田重夫: 環境適応のためのチャンキングメカニズム, 情報処理学会人工知能研究会, 74-4 (1991).

付録

(11) 式はつぎのように導出される.

$$f(x_1, x_2, y) = \begin{cases} (x_1 - y)^2 & \text{if } y < x_1 \\ 0 & \text{if } x_1 \leq y \leq x_2 \\ (x_2 - y)^2 & \text{if } x_2 < y \end{cases}$$

において, y が区間 $[q, q+p]$ の一様分布に従うとき $0 < w < p$ であるような w についての $f(x, x+w, y)$ の平均値を $g(x, w, q, p)$ とおくと,

$$\begin{aligned} g(x, w, q, p) &= \frac{1}{p} \left(\int_0^{x-q} x^2 dx + \int_0^{q+p-x-w} x^2 dx \right) \\ &= \frac{1}{3p} \{(x-q)^3 + (q+p-x-w)^3\}. \end{aligned}$$

x が区間 $[q, q+p-w]$ の一様分布に従うときの $g(x, w, q, p)$ の平均値を $h(w, q, p)$ とおくと,

$$\begin{aligned} h(w, q, p) &= \frac{1}{p-w} \int_q^{q+p-w} g(x, w, q, p) dx \\ &= \frac{1}{6p} (p^3 - 3p^2w + 3pw^2 - w^3). \end{aligned}$$

よって,

$$\begin{aligned} B(w, p) &= h(0, q, p) - h(w, q, p) \\ &= \frac{w^3}{6p} - \frac{w^2}{2} + \frac{wp}{2} \end{aligned}$$

とすれば, $f(x, x+w, p) + B(w, p)$ の平均値は, 任意の w について $p^2/6$ となる. ■