

空間分割を用いた識別と非線形写像の学習

(2) データ空間の再帰的分割に基づく非線形写像学習

— 回帰木の今昔 —

中村 恭之

和歌山大学システム工学部
ntakayuk@vrl.sys.wakayama-u.ac.jp

和田 俊和

和歌山大学システム工学部
twada@ieee.org

■ 非線形写像学習の必要性

ロボットなどの実環境で動作する知的システムは、そのシステムが置かれた環境に適応し、適切な行動を選択する能力が不可欠である。この環境適応能力は、センサ情報と行動を対応付けることを意味し、学習と呼ばれる。前回取り上げた識別における学習問題は、入力ベクトルに対して離散クラスラベルを対応付ける問題であったが、今回取り上げる問題は連続的な入力ベクトルを連続的な出力ベクトルに対応付ける、より一般化された問題である。

このような非線形写像を推定する手法は、画像情報からその解釈への写像を求めるコンピュータビジョンの問題、あるいは、連続特徴ベクトルから離散クラスラベルへの写像を求めるパターン認識の問題、「どういう状態」で「どういう行動」といった過去の経験を基にして状態 → 行動の写像を学習するロボットの行動学習問題など、数多くの場面で必要とされている根本的技術である。

■ 非線形写像学習法のこれまで

これまでに提案されてきた非線形写像の学習法は、以下のように、Lazy 学習法と Eager 学習法に大別することができる。

• Lazy 学習法

- 事例として与えられた入力と出力の組をそのまま記憶しておき、新たな入力に対しては、記憶しているデータの中から類似しているデータを検索して、それらのデータから出力値を計算する方法。たとえば、 k -近傍 (k -NN) 法など。

- 入力空間内の部分領域を k -NN 法によって求め、その部分領域に含まれている学習用データを基に回帰分析を行い写像を推定する方法。この方法では、学習用データをすべて記憶しておく必要がある。たとえば、Locally Weighted Regression (LWR) 法など。

• Eager 学習法

- 入力空間全体を定義域とする複数の関数を合成することによって写像を推定する方法で、複数の関数の合成度合いを決めるパラメータを学習データをもとに求める方法。たとえば、ニューラルネットワーク (NN) 法、Group Method of Data Handling (GMDH) 法、RBF ネットワーク (RBFnet) 法や NGnet 法など。
- 入力空間を再帰的に分割して、その分割を木構造で表し、木の終端ノードに写像先の値を格納しておく方法。たとえば、回帰木 (Regression Tree) など。

これまでの非線形写像学習法にはそれぞれ長所や短所があるもののこれらの手法の中で最も潜在的な能力が高いものは回帰木であると我々は考えている。その理由は以下の通りである。

- k -NN 法では、入出力関係を事例の集合のみで表し、複雑なモデルを用いていないため、事例を記憶するだけで特に学習は必要ないという長所がある一方で、
 - 入力次元数が大きい場合、複数の類似事例を検索する k 最近傍探索処理に多くの時間を費やし写像計算が高速に行えない。
 - 事例データをそのまま記憶するため、大量のメモリを消費してしまう。
- NN, GMDH, RBFnet, NGnet 法では、連続な写像関数が得られるため、ビジュアルサーボなど微分計算を伴う解析に適するという長所がある一方で、
 - ネットワークの構造を決定するためのパラメータを

などの欠点がある。

- NN, GMDH, RBFnet, NGnet 法では、連続な写像関数が得られるため、ビジュアルサーボなど微分計算を伴う解析に適するという長所がある一方で、
 - ネットワークの構造を決定するためのパラメータを

あらかじめ設定しなければならず、複雑な写像を学習する場合、ネットワークの初期重み係数値を適切に設定しないと学習が収束しないことがある。

- 個々のデータに対する写像の推定精度が保証できない。

という欠点がある。

これらに対して回帰木は、

- 原理的に任意の精度で関数近似を行うことができ、学習は必ず収束する。
 - 基本的に関数近似精度を与えるだけで学習が行え、複雑なパラメータの初期値調整の問題はない。
 - 近似計算を行う際には、二分探索木による探索と、簡単な計算しか必要としないため、近似計算はきわめて高速である。
 - 得られた木構造は IF-THEN ルールに変換することができ、ルールや知識発見などの用途に利用できる。
- などの利点がある。一般的な回帰木では、連続な写像関数が得られないという欠点があるが、この欠点は平滑化処理などを行うことで簡単に解消することができる。このように、回帰木には克服困難な著しい欠点は見当たらない。

以下では、この回帰木を中心としたこれまでの研究の流れを概観し、それを一般化しコンピュータビジョン、パターン認識やロボットの行動学習の問題に利用可能なレベルにまで引き上げた、Partially Linear Mapping (PaLM)-tree アルゴリズムについて説明し、最後にこのアルゴリズムをさまざまな問題へ適用した事例について紹介する。

■ 回帰木による非線形写像学習

回帰木は、与えられた入出力データに内在する写像を学習するために、入力空間を複数の部分領域（超直方体）に分割し、各部分領域における関数近似を行う手法である。

このようなアイデアに基づいて回帰を行おうとする試みの歴史は比較的早く、1960年代に Morgan と Sonquist によって Automatic Interaction Detector (AID) アルゴリズム¹⁾が初めて提案された。AID アルゴリズムは、入力空間のある座標軸に垂直な超平面によって入力空間を分割して2つの部分領域を生成し、各部分領域に対応する出力値を定数値で表現する。このとき、左右の部分領域における出力の推定値の2乗誤差の総和が最小になるような分割位置を求める。このようにして空間分割を進めていき、2乗誤差の総和の減少率がある指定値より

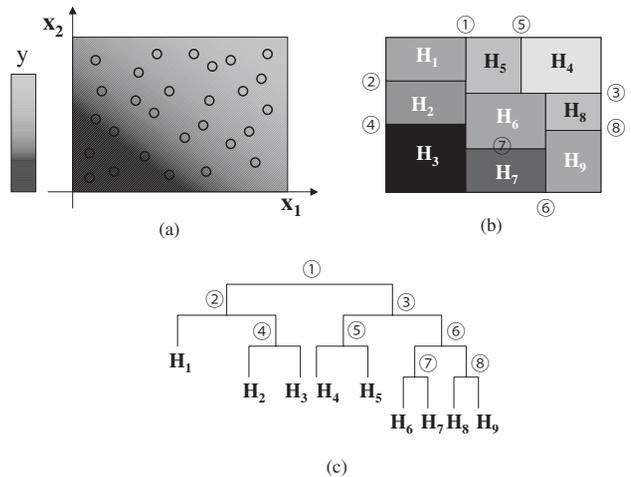


図-1 回帰木による関数近似

も小さくなるか、部分領域の大きさが非常に小さくなれば、入力空間の分割操作を停止する。このような分割方法が原因で、部分領域の大きさが小さすぎたり、大きすぎたりして、over-fitting したり under-fitting したりするという問題があった。

そこで、AID アルゴリズムの入力空間の分割操作を二分木を使って表現し、さらに空間分割の基準を改良したアルゴリズムが1980年代に Breiman らによって提案された。その手法が Classification and Regression Tree (CART)²⁾である。そして、この研究以後、回帰木という名前が普及していった。

図-1は2次元の入力から1次元出力への写像を求めた問題を例とした回帰木概念図である。同図(a)の学習データは位置 $\mathbf{x} = (x_1, x_2)^T$ が入力であり、その点での濃度が出力 y を表している。このとき、回帰木は同図(b)のようになる。すなわち、入力空間を矩形領域に分割し、個々の領域内の近似値を単一の値として保持しておくというものである。入力空間の分割は、同図(c)のような木構造で表現することができ、これを二分探索木として用いることにより、近似計算を行う際に入力されたデータがどの矩形領域内に入ってきたのかを高速に探索できる。先に述べた矩形領域内の近似値は、この二分探索木の葉（以下では、終端ノードと呼ぶ）に格納される。

ここで、従来までの回帰木を用いた非線形写像の学習問題について定式化してみる。この問題は、 p 次元の入力ベクトル $\mathbf{x}_k = (x_{1k}, x_{2k}, \dots, x_{pk})^T$ とスカラ値の出力 y_k の組が n 個 ($k = 1 \sim n$) 与えられたときに、入力ベクトル \mathbf{x} と出力値 y の間の写像関係 $F: R^p \mapsto R^1$ を推定する問題である。ここで、 p 次元ベクトル空間を R^p と表している。

回帰木では、以下に示すように、入力空間内の部分領

域 H_j ごとに回帰関数 $g_j(\mathbf{x})$ を切り替えて入力変数から出力変数への全体的な写像 $F: R^p \mapsto R^1$ を求める.

$$\hat{y} = \sum_{j=1}^{M'} g_j(\mathbf{x}) I_{H_j}(\mathbf{x})$$

ここで, H_1, \dots, H_M は, 入力空間 R^p 内の部分領域を表す. ただし,

$$R^p = \bigcup_{j=1}^M H_j, \text{ where } H_k \cap H_j = \begin{cases} \phi & j \neq k \\ H_j & j = k \end{cases}$$

である. また, $I_{H_j}(\mathbf{x})$ は,

$$I_{H_j}(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in H_j \\ 0 & \mathbf{x} \notin H_j \end{cases}$$

となるような関数である. また, 各部分領域 H_j を木構造におけるノードに対応させることによって, 入力空間の分割を木構造によって表現する. 最終的に, 木のすべての終端ノードに回帰関数が割り当てられる.

回帰木を用いた写像学習には以下のような2つの特徴がある.

- (1) 値域に相当する入力空間を分割して, 値域全体に対する写像の複雑さを軽減させることによって, 各部分領域における回帰関数のパラメータの次数を低くすることが可能になり, 推定された写像関係を解釈したり, 統計的な性質を調査することが可能になる.
- (2) 入力変数間の相関関係も回帰木の木構造に直接的に反映されるので, 入力変数間の相互関係も簡単に解釈することが可能になる.

一方で, 回帰木を構築する際には,

- (1) 回帰関数として何を使用するのか?
- (2) 何を基準として分割判定を行い, どこで空間を分割するのか?
- (3) 分割境界付近で出力の推定値が不連続になることに対してどのように対処するか?
- (4) 汎化能力向上のために分割停止条件を, どう設定するのか?

という問題が浮上してくる.

問題(1)に対しては, 回帰木の初期の研究の CART では, 回帰関数として, 出力の平均値に相当する定数値を割り当てていた. しかし, 回帰木の終端ノードに定数値を割り当てるだけでは, 木のサイズが大きくなりすぎ, また近似も階段状になってしまうという問題がある. この問題を解決するため, Quinlan の M5³⁾ や Karalic の RETIS⁴⁾ では, 各部分領域に入力変数に関する線形関数

を割り当てた. これにより, 1つの線形関数が当てはめられる部分領域のサイズが大きくなるため, 推定される写像の汎化能力が高くなる. さらに, 木の高さが平均的に低くなるため回帰木における二分探索の計算が高速化される. 部分領域のサイズをより大きくするために, 回帰木の終端ノードに入力変数に関する高次多項式を割り当てたもの, たとえば Chaudhuri らの SUPPORT⁵⁾ のようなものも提案されている.

問題(2)に対しては, 一般的な回帰問題を解く場合と同様に, ある部分領域 t についての分割判定の基準として, 以下のような平均2乗誤差 $Err(t)$ を使用することが多い.

$$Err(t) = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_i - \hat{y}_i)^2$$

Karalic らの研究では, この $Err(t)$ に基づき, 入力空間のどの次元 i ($i = 1 \sim p$) のどの位置 $x_i = s$ で分割するのかを決定していた. あるノード t における部分領域のある次元 i の位置 $x_i = s$ において2分割したとして, 生成される2つの部分領域を, それぞれ t_L, t_R と表す. このとき, 以下のような基準値 $\Delta Err(s, t)$ が最大になる分割を見つけるという方法である.

$$\Delta Err(s, t) = Err(t) - \{P_L \times Err(t_L) + P_R \times Err(t_R)\}$$

ここで, P_L, P_R はそれぞれ元の部分領域に存在していたデータが, 分割された部分領域にどのように振り分けられたかを表す割合である.

しかしながら, このような方法では分割次元や分割位置を総当り的に探索して求める必要があるため, 入力が高次元ベクトルの場合には計算量が膨大になるという問題がある. そこで, このような問題を取り扱うためにさまざまな手法が考えられてきた.

たとえば, Chaudhuri らは統計的検定処理に基づいて分割次元と分割位置を決める方法を提案した. まず, 回帰関数として定数値, 線形関数, 高次多項式などを入力データに当てはめて, 正の残差と負の残差を持つデータに分類する. このようにして分類された入力データに関して, 入力変数の各次元ごとに, 正と負の残差を持つ入力データの平均値をそれぞれ計算し, それらの平均値の差に統計的に有意な差があるかどうかを Student の t 検定により最も有意な差がない次元(平均値の差が最小になる次元)を求めて, その次元に関する正と負の残差を持つデータのそれぞれの平均値の midpoint を分割位置とした. しかしながら, 残差の正負により分割位置を決める方法が良いかどうかの根拠は明らかではない. この研究の改良版ともいえる Loh の GUIDE⁶⁾ が提案されたが, その

中でも χ^2 乗検定を用いた同様の処理によって分割次元を決定し、分割位置は探索によって求める方法や先述した平均値に基づく方法や、中央値に基づく方法などが試されている。

また、Dobra らの SECRET⁷⁾ では、EM アルゴリズムを用いて入力データを2つのクラスに分類し、2つのクラスに分割された入力データをもとにクラス境界を表す超平面を線形判別分析により求めて、この超平面を分割位置として決定している。しかしながら、分割位置の決定に EM アルゴリズムを用いているために分類結果に初期値依存性があるという問題があり、さらに、入力データ数が多い場合や入力変数の次元数が大きい場合などには、EM アルゴリズム処理にかかる時間が長くなるという問題もある。

問題 (3) に対しては、Chaudhuri ら⁵⁾ は、入力空間を一度分割した後に、各部分領域について、部分領域を拡張した領域に対して再度回帰関数を当てはめ、重み関数を導入することによって分割境界付近で関数値を連続的に変化させる方法を提案している。Torgo⁸⁾ も、分割した部分領域の近傍を定義するカーネル関数を導入して同様の処理に基づいて分割境界付近においても連続的に出力値を生成する方法を提案している。しかしながら、これらの手法では、各部分領域に関してどの程度まで拡張させたらよいかをあらかじめ決めなければならないなど、調整しなければならないパラメータが多く存在する。

問題 (4) に対しては、一般的には決定木の枝刈り処理と同様の操作を行うことによって対処するものが多い。木の生成過程における分割判定処理の際に、データを分割することによる平均2乗誤差の減少度などの情報から、ある部分空間の分割が推定精度の向上に貢献するかどうかを事前に判定して、入力空間の分割を途中で終了させる方法(事前枝刈り法)や、部分領域に含まれるデータが入力次元数+1程度になるまで細かい分割を行って回帰木を生成した後に過学習の起こっている部分木を削除していく方法(事後枝刈り法)などがある。

しかし、いまだに決め手となる方法は存在していないのが実情である。

この問題に対して Breiman⁹⁾ は、入出力データをランダムサンプリングすることにより、数個のグループに分類して、それぞれのグループに存在する入出力データをもとに回帰木を構成し、複数の回帰木からの出力の重みつき平均値を最終的な推定値とすることによって、回帰木の汎化能力を向上させる手法を提案している。

■ PaLM-tree

これまでの回帰木の研究を概観すると、

- 回帰木の計算ではデータからのルール発見を主な目的としていたため、一般にルールが複雑化する多次元ベクトル空間を値域とする拡張はほとんど検討されてこなかった。
- 入力空間の分割方法が多様であり、簡便かつ信頼性の高い手法が存在しない。

という問題があると言える。このため、回帰木が、コンピュータビジョン・パターン認識・ロボット制御などの分野における用途に本格的に用いることができるまでには至っていない。

そこで、これまでの回帰木の問題点を解消するため、我々は次の2点を拡張した回帰木を提案した¹⁰⁾。

- 出力ベクトルを多次元に拡張し、 p 次元から q 次元への一般的な写像 $F: R^p \rightarrow R^q$ を取り扱えるようにする。
- 分割位置の決定は総当り的に探索することなく、Split-and-Merge 戦略を用いる。

これらの拡張を施した回帰木を区分線形写像木 (Partially Linear Mapping tree: 以下 PaLM-tree) と呼んでいる。

前者の拡張により、先述した一般的な回帰木の問題(1)への、後者の拡張により、問題(2)、(4)への解決策を示している。出力次元を多次元ベクトルとすること自体に技術的な困難さはないが、この拡張により応用範囲がかなり広がる。たとえば、入力空間に出力を多様体として埋め込むパラメトリック固有空間法で扱えるのは $p \geq q$ の場合に限られていたが、PaLM-tree では、 $p < q$ の場合も扱えるため、 $(x, y) \mapsto (r, g, b)$ のように位置から色への写像を求める画像の領域分割の問題などにもそのまま適用できる。また、Split-and-Merge 戦略を採用することにより、学習時間の短縮と写像学習の汎化能力向上を同時に達成している。

分割位置決定方法

分割位置決定の問題に対しては、Horowitz と Pavlidis¹¹⁾ の Split-and-Merge による画像の領域分割法の根底にある考え方、すなわち分割位置を厳密に決定するのではなく、とりあえず条件を満足するまで分割を行い、後から条件を満足する範囲内で領域をまとめれば、条件を満足する最大の領域が得られるという考え方を採用している。

PaLM-tree では、前回の解説で説明した Sliding-mid-

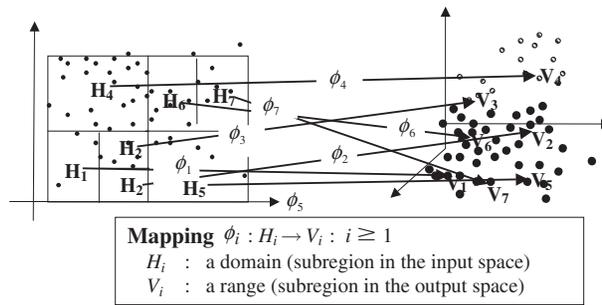


図-2 PaLM-treeによる関数近似

point split rule とは異なる median of max split と呼ばれる空間分割法を用いる。すなわち、各次元についてのデータの分布の幅を調べて、分布の幅が最長の次元を分割する軸として選択し、その分割軸に射影したデータの中央値を分割位置とする方法である。これにより、分割によってデータ数の偏りができるだけ起きないようにしている。

この分割によって得られた領域内のデータに対して回帰計算を行い、回帰誤差が指定された範囲内に収まるまで分割を再帰的に行う (図-2 参照)。分割が進むと部分領域内のデータ数が減少するが、このとき、データ数が入力空間の次元数以下になったり、データ数が少なすぎて入力変数間に非常に強い相関関係が出てきてしまったりして、一般的な回帰計算を行うことができなくなる。この問題に対処するために、PaLM-tree では入力データを特異値分解 (SVD) して一般逆行列を求めて回帰計算を行う。

この分割が終了した後に、今度は隣接する部分領域内のデータを統合して、共通の回帰係数で写像が推定できるかどうか、すなわち Merge できるか否かを判定するために回帰計算を行う。この計算では Merge によって新たな部分領域を生成するのではなく、各部分領域における回帰係数を共有化するようにしている。これにより、分割境界付近での写像推定値が不連続になる傾向を抑えている。

回帰計算

入力ベクトル $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})^T$ に対応する出力ベクトルを $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iq})^T$ とする。これらのデータ組が n 個得られたとき、全データは、

$$\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n], \mathbf{Y} = [\mathbf{y}_1 \cdots \mathbf{y}_n]$$

という 2 つの行列で表現することができる。

\mathbf{B} を $p \times q$ の係数行列としたとき、 \mathbf{x}_i に対する回帰式は $\mathbf{B}(\mathbf{x}_i - \bar{\mathbf{x}}) + \bar{\mathbf{y}}$ と表すことができる。ただし、 $\bar{\mathbf{x}}, \bar{\mathbf{y}}$ はそれぞれ $\mathbf{x}_i, \mathbf{y}_i, (i = 1 \sim n)$ の平均ベクトルである。ここで、

$\mathbf{X}' = [\mathbf{x}_1 - \bar{\mathbf{x}} \cdots \mathbf{x}_n - \bar{\mathbf{x}}], \mathbf{Y}' = [\mathbf{y}_1 - \bar{\mathbf{y}} \cdots \mathbf{y}_n - \bar{\mathbf{y}}]$ とし、誤差ベクトル $\mathbf{e}_i = \mathbf{y}_i - \mathbf{B}\mathbf{x}_i$ を導入することによって、 $\mathbf{Y}' = \mathbf{B}\mathbf{X}' + \mathbf{E}$ という回帰式が得られる。ただし、 $\mathbf{E} = [\mathbf{e}_1 \cdots \mathbf{e}_n]$ である。この式において、 \mathbf{X}' を特異値分解した行列で置き換えると $\mathbf{Y}' = \mathbf{B}(\mathbf{U}\mathbf{D}\mathbf{V}^T) + \mathbf{E}$ という式が得られるため、 $\mathbf{B} = \mathbf{Y}'\mathbf{V}\mathbf{D}^{-1}\mathbf{U}$ と回帰係数を推定できる。この回帰係数の計算法は主成分回帰分析と呼ばれる方法であり、独立なデータ数が入力空間の次元数以下でも破綻することなく実行することができることが知られている。また、特異値分解によって計算された最大特異値と最小特異値の比が大きな値を持つ次元に関して、その特異値を 0 にすることにより、特徴選択を行うことが可能になるため、このような方法を採用している。

回帰誤差の評価

ここでは、PaLM-tree において部分領域を分割するかどうかを判定する際の誤差評価の方式について説明する。しかし、この評価は適用する問題に依存して決定すべき内容であり、以下に述べる方式が唯一ではないことに注意されたい。

出力ベクトルを多次元に拡張し、 p 次元から q 次元への写像 $F : R^p \rightarrow R^q$ 一般を取り扱えるようにするためには、単純にスカラー値を出力する回帰木を組み合わせるだけでは実現できない。そのため、多次元ベクトルを出力するときの誤差評価を考え直さなければならない。単純に部分領域内での回帰誤差の平均値 $\frac{1}{n} \sum_i \|\mathbf{y}_i - \mathbf{B}\mathbf{x}_i\|$ を用

いた場合には、次に示すような不都合が生じる。 \mathbf{y}_i は q 次元ベクトルであり、1 つの部分領域内部で、ある成分は変動が大きく、ある成分は変動が小さいということが一般的に起こり得る。このため、変動の大きさを考慮した誤差評価を行わなければならない。出力ベクトルを $\mathbf{y}_i = (y_{i1}, y_{i2}, \dots, y_{iq})^T$ としたとき、 y_{ij} の偏差を σ_j

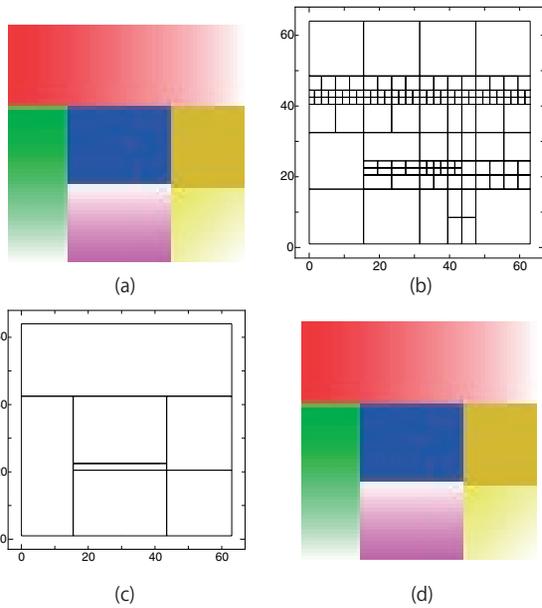


図-3 PaLM-treeによる画像領域分割

と表すものとする。このとき、 $e_j = \sum_i \frac{\|y_{ij} - (Bx_i)_j\|}{\sigma_j}$ とし、 $e = \sum_j e_j$ によって回帰誤差を評価する。これによって、各成分の変動幅の影響を受けにくい分割判定が行える。

PaLM-treeの応用例

PaLM-treeの用途は非常に多岐に渡るが、ここでは画像の領域分割、カメラキャリブレーション、顔の方向推定、そしてロボットの行動学習問題への応用例について述べる。

画像の分割

カラー画像は2次元の画像座標値 (x, y) から3次元の色情報 (r, g, b) への写像であると考えることができる。すなわち、 $R^2 \mapsto R^3$ という写像である。これを学習することは、画像の領域分割を行うことにほかならない。図-3に示した例は、グラデーションをつけた複数の領域からなる画像データに対して領域分割を行った場合の結果である。図-3(a)は原画像、(b)は分割のみを行った場合の部分領域であり、(c)は引き続き統合処理を行った結果である。(d)は最終的に得られた回帰係数から画像を復元した結果である。(b)の領域数は143個であり、(c)では7個まで減少している。このことから、Split-and-Mergeに基づく統合計算の有効性が確認できる。

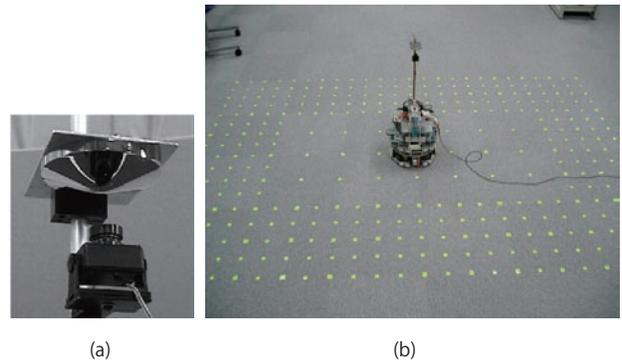


図-4 自由曲面ミラーを用いた全方位カメラと実験環境

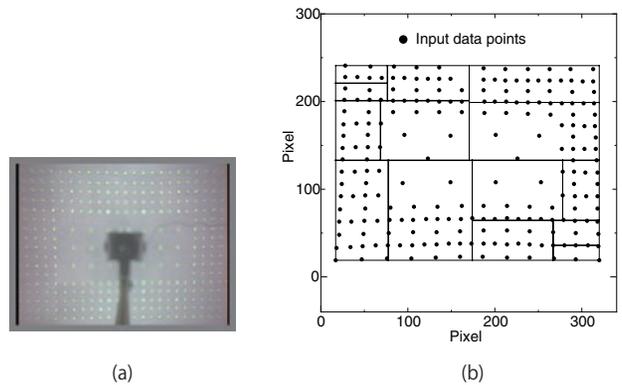


図-5 取得画像とキャリブレーション結果

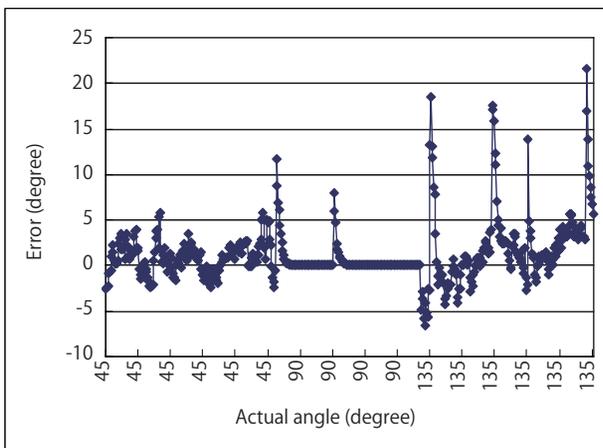
カメラキャリブレーション

歪のあるカメラによって得られた画像を補正する問題は、位置 (x, y) から補正後の位置 (x, y) への写像、すなわち、 $R^2 \mapsto R^2$ という写像としてとらえることができる。ここでは、自由曲面ミラーを用いた全方位カメラ(図-4(a)参照)のカメラ装着位置をずらすことによって歪みを生じさせた画像のキャリブレーションを行った結果を示す。図-4(b)にその撮影風景を示す。このときに撮影された画像を図-5(a)に示す。被写体は平面上に規則的に配置された点群であるが、同図(a)に示す画像には歪みが生じている。この画像上の点群の位置を入力ベクトル、平面上の点群の位置を出力ベクトルとしてPaLM-treeで学習を行い、キャリブレーションを行った結果を同図(b)に示す。この図から、歪んだ点配置が正射影に近い規則的配置に戻されていることが確認できる。

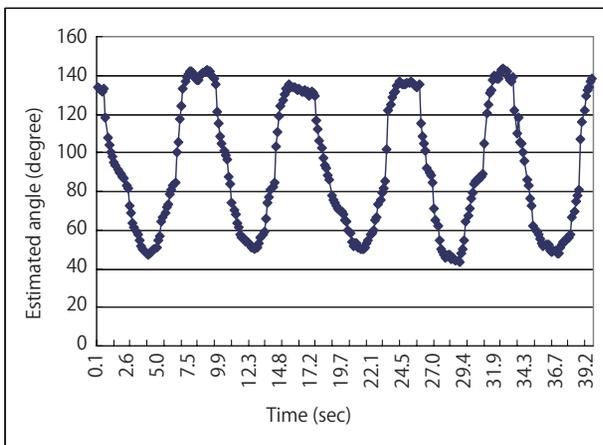
普通の全方位カメラや、広角レンズを付けたカメラにも光学的な歪みが生じており、これらについてもPaLM-treeを用いて、画像の補正が可能である。



(a)



(b)



(c)

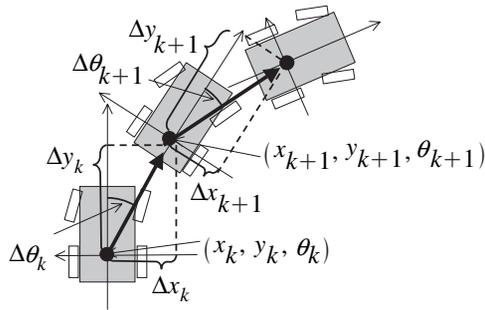
図-6 PaLM-treeによる顔方向推定

複数カメラ画像からの顔方向の推定

住居内で人間の行動を観測・学習して行動意図を推測し、システムから積極的にサービスを提供する能動的な生活支援パートナーを開発する「ゆかりプロジェクト (UKARI: Universal Knowledgeable Architecture for Real-Life appliances)」が、現在、情報通信研究機構が中心となって実施されている¹²⁾。このプロジェクトの中で、行動意図推測の手がかりとして顔方向を推定するシステムが開発されている。

このシステムでは、人物を撮影している複数のカメラにより得られる顔領域の面積から顔方向を推定するために、ある顔方向と顔領域面積の対応を学習させている。あらゆる方向に対応するために、人間を囲むように配置されている4台のカメラを用いる。したがって、4つの顔領域面積の値 (S_1, S_2, S_3, S_4) から顔方向 θ への写像、すなわち、 $R^4 \mapsto R^1$ という写像を学習する。学習は、被験者の体の向きに対し左方向を0度、正面を90度、右方向を180度とする座標系で、顔の向きが、45度、90度、135度の3種類の方向について、各方向に関して約30秒の顔画像を撮影しそのときの顔領域抽出結果を入力として記録した。顔領域の抽出システムは秒間10フレームの速度で処理が行われているので、約300種類の面積値を4台のカメラ分だけ記録したものを学習用のデータとして用いている。

図-6(a)に、4方向のカメラ画像における顔領域抽出結果の例を示す。この図から分かるように、学習用に記録されたデータの中には、顔領域でない物体の面積値も含まれており、この例では、学習用データ内にある程度のノイズも含まれている状況下で写像 $R^4 \mapsto R^1$ を学習しようとしていることになる。なお、学習用のデータを取得している間は、顔方向については、3種類の各方向に固定させている。図-6(b)に、学習後に改めて顔方向を45度、90度、135度とし、4つのカメラより得られる入力画像から顔方向を推定した場合の、各角度と推定値の差を示す。最大で20度程度の誤差があることが分かるが、正面と左方向、右方向の区別は確実にされている。また、顔方向を固定させずに左右に振った場合の方向推定結果を図-6(c)に示す。この場合は、被験者は45度から135度の間を、40秒間で5往復の速度で左右に顔を振っている。この図より、顔振りに追従して、45度から135度の間を約40秒間で5往復している様子が検出できていることが分かる。この例のように、PaLM-treeは、リアルタイムで画像認識を行うシステムに簡単に組み込むことができる。



Input parameters
 $(\Delta x_{k-m}, \Delta y_{k-m}, \Delta \theta_{k-m}, accel_{k-m}, steer_{k-m})$

Output parameters
 $(\Delta x_{k+1}, \Delta y_{k+1}, \Delta \theta_{k+1})$

図-7 移動ロボットの動特性を学習するための入出力データ

車輪型移動ロボットの行動学習

この例では、ロボットの挙動を表す大量の入出力データを蓄積し、これらの事例データを基にロボットの入出力間の相関関係、つまり、ロボットの滑る動きや車体の動特性を考慮に入れた遷移モデルを推定する問題を扱っている。

実際に車輪型移動ロボットを走行させ、その時のロボットの位置・姿勢 (x, y, θ) 、速度・操舵角指令値 $(accel, steer)$ を記録・蓄積する。図-7に示すように、ある時刻 k におけるロボット座標系を、ロボットの2つの後輪の中点を原点、ロボットの車軸方向を x 軸、ロボットの進行方向を y 軸になるように定義し、時刻 $k+1$ におけるロボットの位置・姿勢から、時刻 k における変位量 $(\Delta x_k, \Delta y_k, \Delta \theta_k)$ を計算する。蓄積されているデータに対して同様の計算を繰り返し適用して、各時刻における変位量を計算する。

実際にロボットを走行させると、現在の変位量は過去の状態に依存して変化することが分かる。そこで、現在を含めた過去数時刻分の変位量と動作指令値をまとめて“現在”における入力ベクトル $(\Delta x_{k-m}, \Delta y_{k-m}, \Delta \theta_{k-m}, accel_{k-m}, steer_{k-m})^T : m=0, 1, 2, 3, 4$ とし、時刻 $k+1$ における変化量 $(\Delta x_{k+1}, \Delta y_{k+1}, \Delta \theta_{k+1})^T$ を出力ベクトルとした。したがって、この例では、25次元の入力ベクトルから3次元の出力ベクトルへの写像、すなわち、 $R^{25} \mapsto R^3$ という写像を学習する。

図-8に実験環境を示す。この例では、車輪型移動ロボットとして図示したようなラジコンカーを使用した。天井付近に設置されたカメラにより、ラジコンカー上部に装着された2つの色マーカを画像内で追跡して、時々

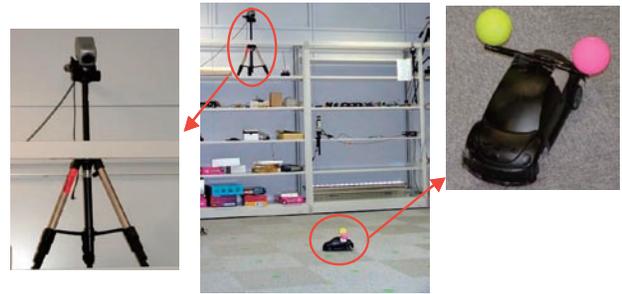
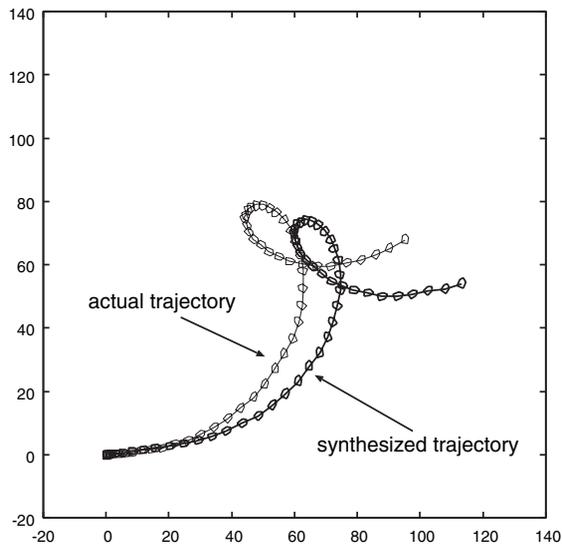


図-8 移動ロボットの動特性を学習するための実験環境

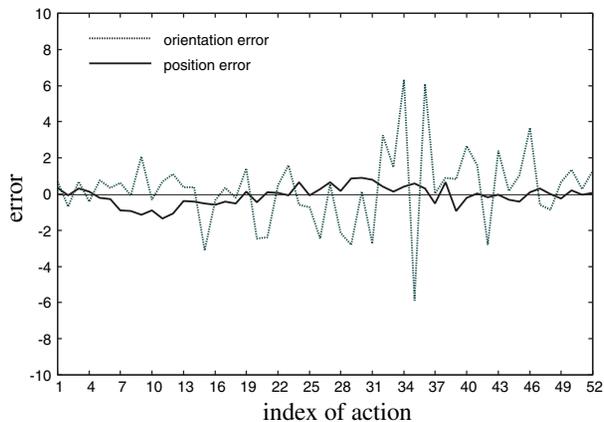
刻々と変化するラジコンカーの位置・姿勢と、そのときに送信されている動作命令を 33msec ごとに記録し、それらのデータを蓄積する。画像中の色マーカの追跡には、最近傍識別器に基づく色ターゲット追跡システムを使用している。カメラで得られた画像座標値を、ラジコンカーが走行している床面を真上から正射影した世界座標系(2次元)に変換するために、ホモグラフィ行列を利用し、ラジコンカーの位置・姿勢を求めている。操舵角指令は右・左ターン・直進の3種類、速度指令は1種類に設定し、合計3種類の動作指令値を用いてラジコンカーを動作させた。このようにして学習用データを約7,000個収集して、PaLM-treeにより、このラジコンカーの状態遷移モデルとして $R^{25} \mapsto R^3$ という写像を学習した。

学習により得られた状態遷移モデルの妥当性を評価するために、状態遷移モデルに基づいて、ある初期状態から最終状態に至る行動系列を分枝限定法を用いて求めて、求められた行動系列によって実際のラジコンカーを動作させたときの軌道と、計算機内でラジコンカーの動きを再現させたときの軌道を比較した。図-9(a)に、その比較結果を示す。

図-9(b)には、求められた行動系列によって実際のラジコンカーに動作させたときの軌道と、求められた状態遷移モデルに基づいて計算した軌道における、1つの動作指令に対する位置変位量・姿勢変位量の差をすべての動作に関して計算した結果のグラフを示す。縦・横軸はそれぞれ、位置変位量・姿勢変位量の差と行動系列内の Index を表す。このグラフを見れば分かるように、すべての動作命令に関して1動作命令あたりの位置・姿勢に関する変位量に関する差は小さく、位置・姿勢の変位量に関する差の平均は $(-0.09, 0.2)$ 、標準偏差は $(0.5, 2.1)$ であり、PaLM-treeにより同定された状態遷移モデルが、実際のラジコンカーの挙動を正確に表現できていることが分かる。したがって、図-9(a)において、提案手法により計算された軌道と実際のラジコンカーの軌道が異なる



(a)



(b)

図-9 PaLM-treeによる移動ロボットの動特性の学習

るのは、局所的な誤差が積算されるためであると考えられる。

■まとめ

非線形写像学習の方法として、従来からの回帰木に関する研究の概要とそれを一般化、高速化した我々の提案する PaLM-tree について述べた。実験結果からも、GMDH, Neural Net, k -NN 法に対する PaLM-tree の優位性は明白であると言える。PaLM-tree では、一般的な回帰木が持つ問題 (3) への解決策は特に示していない。なぜなら、用途によっては分割境界付近で関数値が不連続になっていても問題が生じないこともあるからで、たとえば、画像の領域分割などへの応用では画像の輝度値がそもそも離散値であるので、値が不連続になる部分が

あっても問題にはならない。また、分割境界付近で関数値を連続的に変化させる必要がある場合には、重み関数を導入することによって隣接する領域内の複数の関数値を加算することによって簡単に対処できるからである。PaLM-tree の用途はコンピュータビジョン、制御、計測、パターン認識など非常に多岐にわたり、これからもさまざまな分野への応用を進めていく予定である。

なお、紙面の都合上、回帰木以外の非線形学習法に関する参考文献はどれも有名な手法であるので、参考文献リストを載せていないがお許しいただきたい。

謝辞 PaLM-tree は、和歌山大学システム工学部 Vision and Robotics Laboratory のメンバとの日頃からの議論の中から生まれたものである。PaLM-tree の応用に関する実験データは NICT 佐藤哲氏、和歌山大学システム工学研究科川原輝美氏からいただいた。

参考文献

- 1) Morgan, J. and Sonquist, J.: Problems in the Analysis of Survey Data, and a Proposal, American Statistical Association Journal, Vol.58, pp.415-434 (1963).
- 2) Breiman, L., Friedman, J., Olshen, R. and Stone, C.: Classification and Regression Trees, Wadsworth Int. Group, Belmont, California, USA (1984).
- 3) Quinlan, J. R.: Learning with Continuous Classes, Proc. 5th Australian Joint Conference on Artificial Intelligence, pp.343-348 (1992).
- 4) Karalic, A.: Linear Regression in Regression Tree Leaves, Proc. International School for Synthesis of Expert Knowledge 1992, pp.151-163 (1992).
- 5) Chaudhuri, P., Huang, M. C., Loh, W. Y. and Yao, R.: Piecewise-polynomial Regression Trees, Statistica Sinica, Vol.4, No.1, pp.143-167 (1994).
- 6) Loh, W. Y.: Regression Trees with Unbiased Variable Selection and Interaction Detection, Statistica Sinica, Vol.12, No.2, pp.361-386 (2002).
- 7) Dobra, A. and Gehrke, J.: Secret: A Scalable Linear Regression Tree Algorithm, Proc. 8th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp.481-487 (2002).
- 8) Torgo, L.: Functional Models for Regression Tree Leaves, Proc. 14th International Conference on Machine Learning, Morgan Kaufmann, pp.385-393 (1997).
- 9) Breiman, L.: Bagging Predictors, Machine Learning, Vol.24, No.2, pp.123-140 (1996).
- 10) 中村恭之、加藤丈和、和田俊和：非線形写像学習のための PaLM-tree の提案、日本ロボット学会誌、Vol.23, No.6, pp.732-742 (2005).
- 11) Horowitz, S. and Pavlidis, T.: Picture Segmentation by a Directed Split-and-Merge Procedure, Proc. 2nd International Conference on Pattern Recognition, pp.424-433 (1974).
- 12) 美濃導彦：ゆかりプロジェクトの目的と概要 - UKARI プロジェクト報告 No.1 - , 第 66 回情報処理学会全国大会予稿集, Vol.5, pp.5-8 (2004). (平成 17 年 3 月 3 日受付)



訂正

前回 (8 月号) の解説記事 p.916 左段 2 ~ 3 行目「見つかったプロトタイプを中心とし、入力データを表面に含む超球を考える。」の部分は、「入力データを中心とし、見つかったプロトタイプを表面に含む超球を考える。」の誤りでした。謹んで訂正いたします。