

Shall we start データ駆動地盤工学? ~ベイズ・機械学習を中心として~

第4回 基礎編 (その3) ~予測モデルをつくる~

Basics: Part III – Create a Bayesian-based predictive model

加村 晃良 (かむら あきよし)

東北大学大学院工学研究科 准教授

e-mail: akiyoshi.kamura.a4@tohoku.ac.jp

大竹 雄 (おおたけ ゆう)

東北大学大学院工学研究科 准教授

森口 周二 (もりぐち しゅうじ)

東北大学災害科学国際研究所 准教授

村尾 英彦 (むらお ひでひこ)

(株) 村尾地研 代表取締役

キーワード: ベイズ推定, ガウス過程回帰, 主成分分析, ニューラルネットワーク

1. はじめに

1.1 地盤工学のパラメータ推定は難しい

本講座の第2回, 第3回では, 主にパラメータ推定の問題を取り扱ってきた。本稿 (その3) では, 少し視点を変えて地盤工学分野におけるベイズや機械学習の導入可能性について考えてみたい。その1, その2と同様に改めて下式を見ていただきたい。

$$\mathbf{y} = f(\mathbf{x}) + \varepsilon \quad (1)$$

ここで, \mathbf{y} は観測量ベクトルであり, 地盤の沈下量や変位などを意味する。 $f(\cdot)$ は, 観測量ベクトル \mathbf{y} の生成メカニズムであり一般的に非線形関数である。例えば, 地盤の沈下量や変位を計算する FEM 解析や設計基準に記載されている設計計算式で代替される。 \mathbf{x} は $f(\cdot)$ のパラメータベクトルで, ε は誤差ベクトルである。ここで, その1では, \mathbf{x} , \mathbf{y} , ε が正規分布に従い, $f(\cdot)$ が線形である特殊な例, もしくは線形に近似できる場合を想定した。 \mathbf{x} の推定問題の解析解が示されている。その2は上記の制約を外して, 非正規分布, 非線形関数の場合における \mathbf{x} の推定問題について記載されている。粒子フィルタや MCMC など粒子を利用した方法により \mathbf{x} の事後分布推定問題が解説されている。

さて, 本稿 (その3) の着目は上記と少し異なり, 上記推定問題で“既知”であると仮定されてきた $f(\cdot)$ である。一般的に地盤材料の挙動は複雑なメカニズムに支配されているため, $f(\cdot)$ は複雑化させる必要があり, 高

度な構成則を導入した数値解析等で代替される場合が多い。これにより, パラメータベクトル \mathbf{x} は高次元になり, $f(\cdot)$ は陽に表せない強非線形関数となる。この場合, 第2回で紹介された手法は適用できないし, $f(\cdot)$ の計算コストは大きく, その2で紹介された手法の繰り返し計算に耐えることができない。従って, 地盤工学のパラメータベクトル \mathbf{x} の推定問題を解くのはとても厄介である。

以上の問題意識に基づいて, 本稿では, パラメータベクトル \mathbf{x} の推定や観測量ベクトル \mathbf{y} の将来予測を簡単にする $f(\cdot)$ の作り方 (予測モデルの作り方) に焦点をあてる。“予測モデルの作り方”は, 代理モデル, 応答曲面, サロゲイトモデルなど, 様々に呼称され, 地盤工学分野でも近年活発に研究がなされている。“予測モデルの作り方”には, 極めて多様な機械学習モデルが適用されており, Phoon and Zhang (2022)¹⁾, Zhang et al. (2021)²⁾, Baghban et al. (2021)³⁾ に詳しい。ここでは, そのような手法を網羅するのではなく, “予測モデルの作り方”のコツを分類整理し, その主要な方法について概説することを目的とする。

1.2 $f(\cdot)$ を作り出す

パラメータベクトル \mathbf{x} と観測量ベクトル \mathbf{y} が準備できる場合, そのまま両者を関連づけてしまえば便利である。伝統的には, 多項式や線形基底関数を用いることで両者の非線形関係を合理的に関係づけることがで

きることが知られている。ただし、このとき苦勞するのが「モデルの選択」の問題である。何次関数で近似するのか、どの基底関数群を用いるのか、如何にして過学習を防ぎ、外挿推定を安定させるのか、解析者の不安は尽きない。簡単には、交差検証法 (Cross-validation) によりテスト損失を最小化することで妥当なモデルを選択する機会が多いようである。ただし、地盤工学では十分な教師データを準備できない場合も多いため交差検証法が有効に機能しない場合もある。このようなときには、本講座の中心課題であるベイズの定理に基づくモデル比較が有効に機能する機会が多い。本稿では詳細は述べないが、モデルエビデンスの最大化問題として定式化することで簡単かつ安定的にモデル選択ができる場合がある。

さて、 $f(\cdot)$ を作り出すことを考える場合、やはりニューラルネットワークは強力なツールである。パラメータベクトル \mathbf{x} と観測量ベクトル \mathbf{y} の非線形関係を直接モデル化でき、また、多様なパラメータベクトル \mathbf{x} を設計できる点が魅力の一つである。地盤の沈下量 \mathbf{y} 、地盤のヤング率や強度定数 \mathbf{x} の関係を作ることもでき、もしかしたら、地盤の沈下量 \mathbf{y} 、採取した地盤のコアを撮影した写真をそのまま \mathbf{x} にすることも可能となる。地盤工学が陥りがちな情報の不足や経験工学的要素をニューラルネットは克服してくれる可能性を秘めている。

表-1 本稿における手法の分類

分類	主な手法
$f(\cdot)$ を作り出す	回帰分析 (Polynomial 関数, 線形基底関数), サポートベクターマシン, ニューラルネットワークなど
$f(\cdot)$ の滑らかさを利用する	ガウス過程回帰 (GPR, Gaussian Process Regression), 固有直交分解 (POD, Proper Orthogonal Decomposition), ニューラルネットワークなど

1.3 $f(\cdot)$ の滑らかさを利用する

図-1は、例としてバネにつながれた質点の変位を示すものである。質点変位の時間変化は支配する物理(メカニズム)に応じて滑らかに分布している。例えば、図中の黒点のみが観測され、その間を予測する問題を考えてみよう。この質点のバネや減衰係数がわからな

くても、観測点の間の質点変位は予想できそうである。すなわち、観測量ベクトル $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2\}$ の一部 \mathbf{y}_1 を観測して、それ以外の \mathbf{y}_2 を推定する時、必ずしもパラメータベクトル (バネや減衰) \mathbf{x} を介する必要はないかもしれない。このことが、次章に示すガウス過程回帰のカーネルトリックの直感的理解である。ガウス過程回帰では、観測量ベクトル \mathbf{y} の滑らかさをカーネル関数で構築した共分散行列でモデル化⁴⁾し、固有直交分解では、観測量ベクトル \mathbf{y} から標本の共分散行列とその固有ベクトル計算して上手に利用する⁵⁾。もちろん、ニューラルネットワークも滑らかさの同定に用いることができる有効な手段の一つである。図-1からもわかる通り、観測量ベクトル \mathbf{y} の滑らかさには、観測量を支配する物理 (メカニズム) や支配パラメータの特徴が埋め込まれており、その特徴を丁寧に分析することで観測量の背景にある未知な物理 (メカニズム) を考察できる可能性がある。

これ以降、上記に示したベイズや機械学習導入のコツを具体的に解説する。まず、近年多様な予測問題に適用されているガウス過程回帰と固有直交分解について述べる。その上で、最後に汎用的なツールとして多様な問題に応用できるニューラルネットワークの基礎について解説することにする。

なお、本稿では前回までの講座に現れた数式 (または図) を参照する際は、講座の回番号を式番号 (または図番号) に付し、例えば「式(1.10)」のようにして「第1回の講座に現れた式(10)」を表すものとする。

2. ガウス過程回帰

2.1 なぜガウス過程回帰なのか

式(1)で示した \mathbf{y} について、分布すなわち $f(\cdot)$ の滑らかさを利用する方法としてガウス過程回帰 (GPR, Gaussian Process Regression) がある。ガウス過程回帰は、その名のとおりにガウス過程の理論に基づく回帰分析手法の一つである。確率 $p(\mathbf{y})$ がガウス分布 (正規分布) に従うと仮定することで、分布の予測とともにその不確実性 (信頼区間) を提供できるというメリットがある。ここでは、線形回帰との比較から、ガウス過程回帰の仕組みについて、導入するモチベーションと共に概説する。

例えば、一次元の入力 x について、三つの基底関数を成分に持つ特徴ベクトル $\phi(x) = (1, x, x^2)^T$ を考えると、 x の 2 次関数 $y = w_0 + w_1x + w_2x^2$ は重みを $\mathbf{w} = (w_0, w_1, w_2)^T$ とする線形回帰モデルとして以下のように表すことができる。

$$y = \mathbf{w}^T \phi(x) \quad (2)$$

ここでの線形回帰とは、式(2)のように線形方程式で表される形式をいう。しかし、次元の呪いと言われるように、入力 x の次元が次元を超えると計算コストが膨大となり、非実用的になってしまう。

結論から言えば、この問題の解決法は、重み \mathbf{w} の期待値をとって積分消去することである。簡便のため、予測値 y に誤差がないと仮定し、二次元以上の入力 \mathbf{x} の特徴ベクトル $\phi(\mathbf{x})$ をまとめて Φ と表せば $\mathbf{y} = \Phi \mathbf{w}$ と書ける。重み \mathbf{w} が平均 $\mathbf{0}$ で分散が $\lambda^2 \mathbf{I}$ のガウス分布から生成される ($\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \lambda^2 \mathbf{I})$) とすれば、これはガウス分布に従うベクトル \mathbf{w} を Φ で線形変換したものと解釈できる。このとき、 $\mathbf{y} = \Phi \mathbf{w}$ もガウス分布に従うので、期待値をとると以下ようになる (証明略)。

$$E[\mathbf{y}] = E[\Phi \mathbf{w}] = \Phi E[\mathbf{w}] = \mathbf{0} \quad (3)$$

また、分散共分散行列 Σ は、 $E[\mathbf{w}\mathbf{w}^T] = \lambda^2 \mathbf{I}$ を利用すると、以下ようになる。

$$\begin{aligned} \Sigma &= Cov[\mathbf{y}] = E[\mathbf{y}\mathbf{y}^T] - E[\mathbf{y}]E[\mathbf{y}]^T \\ &= E[(\Phi \mathbf{w})(\Phi \mathbf{w})^T] - E[\mathbf{y}]E[\mathbf{y}]^T \\ &= \Phi E[\mathbf{w}\mathbf{w}^T] \Phi^T \\ &= \lambda^2 \Phi \Phi^T \end{aligned} \quad (4)$$

よって、 \mathbf{y} の分布は、式(6)に示す多変量ガウス分布として得られる。

$$\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \lambda^2 \Phi \Phi^T) \quad (5)$$

この式(5)は、期待値がとられて重み \mathbf{w} が消えている。つまり、 \mathbf{x} の次元が高くとも、対応する重み \mathbf{w} を求める必要がなく、 \mathbf{y} の分布は、データ数 N に依存する共分散 $\lambda^2 \Phi \Phi^T$ で得ることができる。 N 個の入力 \mathbf{x} について、出力 \mathbf{y} の同時分布 $p(\mathbf{y})$ が多変量ガウス分布に従うとき、 \mathbf{x} と \mathbf{y} の関係はガウス過程に従うという。

線形回帰では、一般に最小二乗法により重み \mathbf{w} が決定される。一方、ガウス過程回帰では、後述するカーネル関数によって、正規分布を仮定した $p(\mathbf{y})$ の分散共分散行列を直接的に与えることで、重み \mathbf{w} を計算す

る必要がなくなる。

観測データ \mathbf{y} について、あらかじめ平均値を引いて平均 $\mathbf{0}$ とした場合のガウス過程は式(5)に示したとおりであり、これは \mathbf{y} の分布が分散共分散行列 Σ の (n, n') 成分である $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_{n'})$ だけで定められることを意味している。ここで、 \mathbf{x}_n と表記したのは、既に (n, n') 成分に値が入っているためである。この $\phi(\mathbf{x})$ を明示的に求める必要はなく、 $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_{n'})$ の結果だけわかればよい。 $\phi(\mathbf{x}_n)^T \phi(\mathbf{x}_{n'})$ の値を与える関数を \mathbf{x}_n と $\mathbf{x}_{n'}$ のカーネル関数といい、以下のように表す。

$$\kappa(\mathbf{x}_n, \mathbf{x}_{n'}) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_{n'}) \quad (6)$$

これは、特徴ベクトルを書き下して明示的に求めなくとも、カーネル関数によって分散共分散行列を直接的に与えることで、 $p(\mathbf{y})$ が求められることを意味する。このようにカーネル関数だけで内積を計算するテクニックをカーネルトリックといい、これによりいわゆる次元の呪いから解放されることとなる。

カーネル関数には様々な種類があるが、よく使われるのは、動径基底関数カーネル (RBF カーネル) であり、入力 \mathbf{x} について以下のように定義される。

$$\kappa(\mathbf{x}, \mathbf{x}') = \theta_1 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\theta_2}\right) \quad \theta_1, \theta_2 \in \mathbb{R} \quad (7)$$

ここに、 θ_1, θ_2 は関数形を決めるパラメータである。

なお、ここまでは簡便のため予測値に誤差が含まれないと仮定して式を示してきたが、実際はそうではない。データ数 $n = 1, \dots, N$ として、式(1)と同様に、

$y_n = f(x_n) + \varepsilon_n$ を考える。誤差 ε_n は平均 0 、分散 σ^2 のガウス分布 $\varepsilon_n \sim \mathcal{N}(0, \sigma^2)$ に従うとする。このとき、

$$\mathbf{y} = (y_1, y_2, \dots, y_N)^T \text{ の確率分布は次のように書ける。} \quad p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2 \mathbf{I}) \quad (8)$$

ここで、 \mathbf{f} は冒頭で述べた $f(\cdot)$ のベクトル表示を意味しており、 $\mathbf{f} = (f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_N))$ である。このとき、 $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ が与えられた後の分布は、

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma + \sigma^2 \mathbf{I}) \quad (9)$$

となる。この式(9)は、 \mathbf{y} がガウス過程に従い、カーネル関数から定められる分散共分散行列 Σ の対角要素に分散 σ^2 を足したものとして表現されることを意味する。この場合、 \mathbf{y} はカーネル関数を以下のように置いたガウス過程に従う。

$$\kappa'(\mathbf{x}_n, \mathbf{x}_{n'}) = \kappa(\mathbf{x}_n, \mathbf{x}_{n'}) + \sigma^2 \delta(n, n') \quad (10)$$

ここで、 $\delta(n, n')$ は、 $n = n'$ のときに1、それ以外るときには0を返す関数である。

最後に、具体的にどのように回帰問題を解くか、すなわち未知の y^* (補間したい場所の y) をどのようにガウス過程で求めるのかを概説する。入力 \mathbf{x} と出力 \mathbf{y} に関する N 個のペアに対して、物理(メカニズム)に応じて滑らかに分布するモデルが適用できるならば、 $\mathbf{y} = f(\mathbf{x})$ の関係として表すことができることは、本稿の前提条件とするところである。この関数 $f(\mathbf{x})$ が前述のとおり平均0のガウス過程から生成されているとするので、カーネル関数 $\mathbf{K} = \kappa(\mathbf{x}_n, \mathbf{x}_{n'})$ を用いて、 $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ と書ける。このとき、データに含まれていない \mathbf{x}^* で y^* の値を解くことになる。 \mathbf{y} に未知の y^* を含めたものを新たに $\mathbf{y}' = (y_1, y_2, \dots, y_N, y^*)^T$ とし、 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ および \mathbf{x}^* から計算される $(N+1) \times (N+1)$ 次元のカーネル行列 \mathbf{K}' とする。これら全体がガウス過程に従うので、 $\mathbf{y}' \sim \mathcal{N}(\mathbf{0}, \mathbf{K}')$ となることを踏まえると、 \mathbf{y} と y^* の同時分布の式は以下のように表される。

$$\begin{pmatrix} \mathbf{y} \\ y^* \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{pmatrix} \mathbf{K} & \mathbf{k}^* \\ \mathbf{k}^{*T} & k^{**} \end{pmatrix}\right) \quad (11)$$

ここで、 $\mathbf{k}^* = \{k(\mathbf{x}^*, \mathbf{x}_1), \dots, k(\mathbf{x}^*, \mathbf{x}_N)\}^T$ は、補完したい場所の新しい入力 \mathbf{x}^* と既知である \mathbf{x} のカーネル関数の値を並べたベクトルで両者の類似度を意味しており、 $k^{**} = k(\mathbf{x}^*, \mathbf{x}^*)$ は \mathbf{x}^* の自分自身の類似度のような意味を持つ。ガウス過程では、入力が似ていれば出力も似ている、という考え方が根幹にある。

以上の考え方に基づき、式(11)において \mathbf{y} が与えられたときの y^* も平均0のガウス分布に従うので、ガウス過程の予測分布は以下のように得られる。紙面の都合上、式の誘導は省略するが、詳細は参考文献6)を参照されたい。

$$p(y^* | \mathbf{x}^*) = \mathcal{N}(\mathbf{k}^{*T} \mathbf{K}^{-1} \mathbf{y}, k^{**} - \mathbf{k}^{*T} \mathbf{K}^{-1} \mathbf{k}^*) \quad (12)$$

同じように、 y^* の期待値はガウス分布の平均で次のように得られる。

$$E[y^* | \mathbf{x}^*] = \mathbf{k}^{*T} \mathbf{K}^{-1} \mathbf{y} \quad (13)$$

以上のプロセスにより未知の y^* (補間したい場所の y) が求められるが、 \mathbf{x}^* は補間したい場所を表す座標とし

ての扱いとなる。

2.2 線形回帰とガウス過程回帰の比較

紙面の都合上、解法やアルゴリズムを網羅的に示すことは避け、ここでは線形回帰とガウス過程回帰の結果の比較から、ガウス過程回帰の特徴を掴んでもらうこととする。なお、本節における記号表記は、講座第1回から第3回までの基本ルールを踏まえつつ、参考文献6)に極力合わせているので、より深く学びたい方は、そちらを参照されたい。

線形回帰とガウス過程回帰の比較のための問題設定として、1質点・1自由度系の減衰自由振動を考える。運動方程式は、 $m\ddot{u} + c\dot{u} + ku = 0$ であり、 m は質点の質量、 c は減衰定数、 k はばね定数である。これを解析的に解くと、図-1に示すような解が得られる。なお、各パラメータは、1.0から-1.0の範囲に収まるよう適当に調整している。グラフ内のプロットは、線形回帰とガウス過程回帰での参照点(観測点)である。

問題設定をこれまでの説明に対応させると、 $\mathbf{y} = f(\mathbf{a}, t) + \varepsilon$ という観測データにおいて、 \mathbf{a} に運動方程式のパラメータ m, c, k を入れ込み、 t の滑らかさを利用する。つまり、カーネル関数は $\kappa(t, t')$ の形をとり、 t の関係で分散共分散をとっている。第2回基礎編(その1)で触れたモデル選択でいえば、 $\mathbf{y} = \mathbf{M}(\mathbf{a}) + \varepsilon$ と表され、運動方程式が \mathbf{M} に委ねられていると解釈できる。

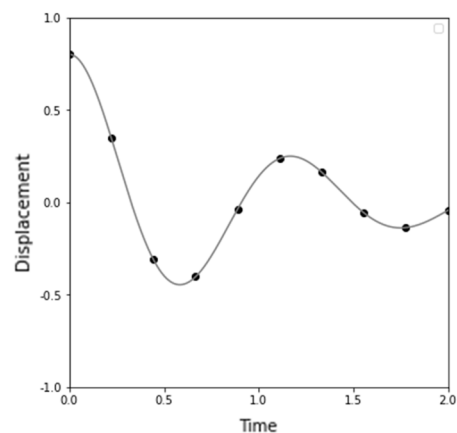


図-1 1質点・1自由度系の減衰自由振動の解とプロット (問題設定)

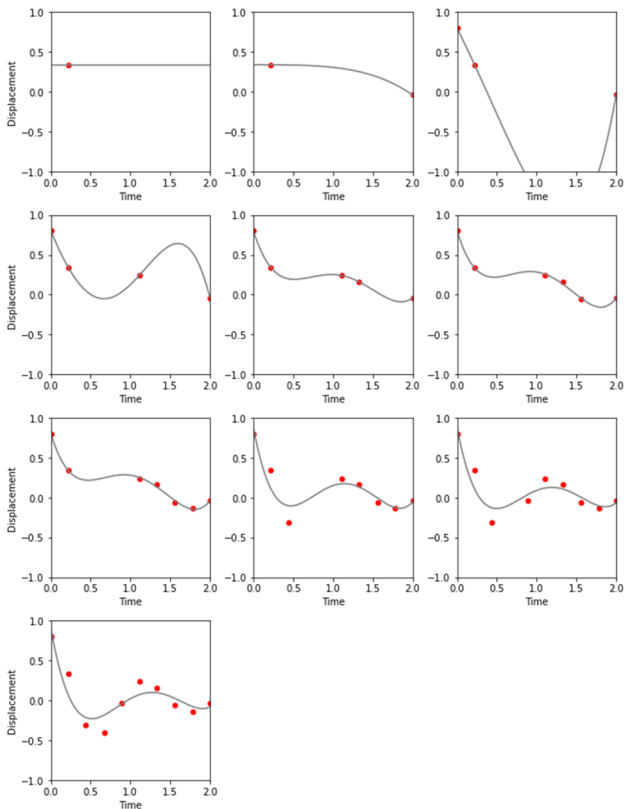


図-2 観測点ごとの線形回帰結果 (4次関数の最小二乗法)

図-2 に 4 次関数を用いた最小二乗法による線形回帰の結果, 図-3 には RBF カーネルを用いたガウス過程回帰の結果を各々示す。両者ともに滑らかな予測を提供するものの, ガウス過程回帰では, 観測点が増えるにつれてより良い予測値を返していることがわかる。また, ガウス過程回帰では, 確率過程がガウス分布に従うため, 予測とともにその不確実性 (信頼区間) を提供していることがわかる。観測点が増えるにつれて, 信頼区間が運動方程式の解析解に収束しており, 予測の信頼性を捉えられるというメリットが理解されよう。

3. モード分解

3.1 モード分解のイメージ

ある物理法則に従って, 入力に対する応答がベクトル (分布) として表現される場合, その中に特徴モード (基底) が存在すると仮定してデータからモードを抽出することで, 入力と応答の関係を効率的に表現することができる。ここでは, モード分解の 1 種の固有直交分解 (POD, Proper Orthogonal Decomposition) をとり上げて説明する。

POD は, データの特徴の抽出や次元削減を目的として使用されることが多く, 特に流体力学の分野での適用が先駆的である⁷⁾。近年では, その適用範囲が大きく拡大しており, 数値解析との融合を経て, 多くの工学的問題に適用されている^(例えば8,9)。POD では, データに対して共通のモードを仮定し, 応答はモードの線形和として表現される。各データの違いはモードの強弱 (線形和の係数の大きさ) として表現される。例えば, 図-4 に示すように, 二次元データ (二つの要素を持つベクトル) のデータ群を考えた場合, 図中の赤と青で示した軸で表現しなおすことで, データをより効率的に表現できる。データの散らばりの度合い (分散) が大きい方向の順に, 第 1 と第 2 の軸を直交するように定義すれば, その方向を示すベクトル \mathbf{u} (図中では \mathbf{u}_1 と \mathbf{u}_2) が特徴モード (基底) として抽出される。また, 各データはモードに係数を掛けた線形和で表現できるため, モードの強弱をコントロールすることで全てのデータを表現できることになる。この操作は, 主成分分析を意味しており, そのために POD のことを主成分分析や特異値分解と呼ぶこともある。

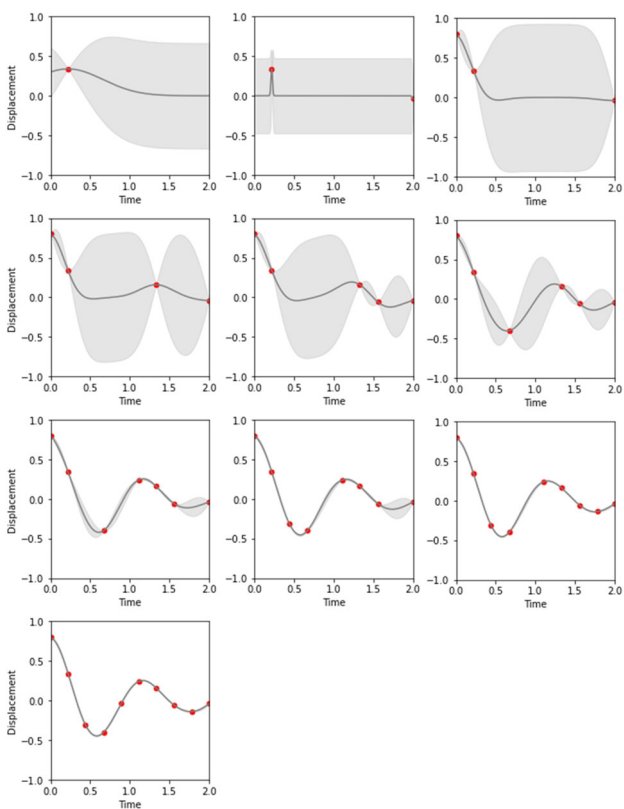


図-3 観測点ごとのガウス過程回帰結果 (RBF カーネル)

上記では二次元データを考えたが、多次元になってもこのイメージは変わらず、データを効率的に説明するためのモード(軸の方向)をデータ自体から抽出し、それらの線形和でデータを表現する。第1, 第2, 第3...とモードが高次になるほどデータに対する寄与度が低くなっていくため、一般的に低次のモードほど物理的意味が強いものとなる。

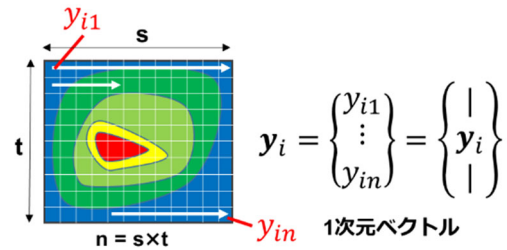


図-6 データのベクトル化

3.2 データ行列

ここでは、POD の具体的な計算に必要なとなるデータ行列について説明する。図-5 に示すように、対象とするデータは、画像データ、空間分布データや時系列データなど、ベクトルとして表現できるものであればデータの種類の限定されない。時系列データは時間方向に数字が並ぶベクトルとしてイメージできるのに対して、画像データや多次元空間(二次元空間や三次元空間)のデータはベクトルとして表現するイメージが難しいかもしれない。しかし、図-6 に示すように、一つのデータ(ここではケース*i*とする)について、それを表現する数字を並べるルールを決めれば、ベクトルを作ることが理解できる。このように、一つのデータはベクトルとして表現されるが、これを列ベクトルとして図-7 のように複数セット(ここでは *m* ケース)分を行方向に並べることで行列の形となる。この行列のことをデータ行列と呼び、これを用いてモード抽出を行う。

$$y_i = \begin{pmatrix} | \\ | \\ | \end{pmatrix} \quad i = 1 \sim m \quad \longrightarrow \quad Y = \begin{pmatrix} | & \dots & | \\ y_1 & \dots & y_m \\ | & \dots & | \end{pmatrix}$$

図-7 データの一次元ベクトル化

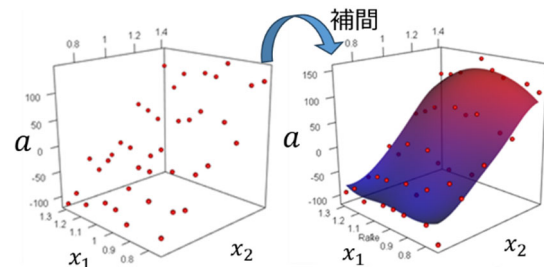


図-8 POD 係数 a_{ij} と入力パラメータの関連付け(補間)

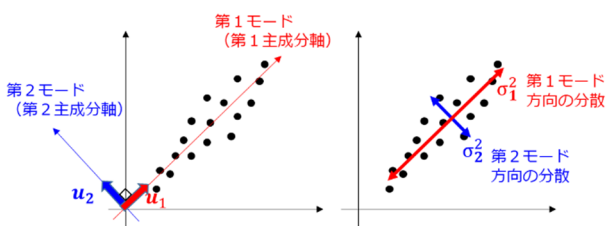


図-4 モード抽出のイメージ

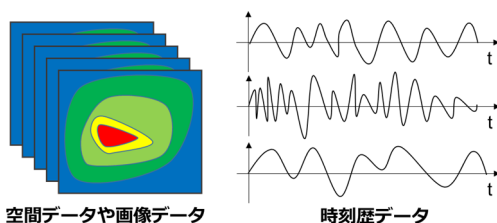


図-5 モード分解の対象となるデータのイメージ

3.3 モードの抽出

先述のように、複数セット (*m* ケース) の多次元データ (*n* 次元) からデータ行列 Y (*n* 行 *m* 列) を作成することができる。モードはデータ行列の共分散行列 ($C = YY^T$) の固有ベクトルとして得られ、特異値を適用すれば下記のように三つの行列に分解できる。

$$Y = U \Sigma V \quad (14)$$

$$= \begin{pmatrix} | & \dots & | \\ u_1 & \dots & u_p \\ | & \dots & | \end{pmatrix} \begin{pmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sqrt{\lambda_p} \end{pmatrix} \begin{pmatrix} - & v_1 & - \\ \dots & \dots & \dots \\ - & v_p & - \end{pmatrix}$$

ここで、 U は共分散行列の固有ベクトルを行方向に並べた行列であり、先述のようにモードが並んでいる行列を意味する。 Σ は共分散行列の固有値の平方根を対角項に並べた行列であり、*p* はゼロではない特異値の数を意味する。また、 V は行列 $C' = Y^T Y$ (共分散行列とは行列の積の順番が逆) の固有ベクトルを並べた行列である。ここで、 ΣV を一つの行列 A として表現すれば

下記のように表現できる。

$$Y = UA = \begin{pmatrix} | & \square & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_p \\ | & \square & | \end{pmatrix} \begin{pmatrix} a_{11} & \square & a_{n1} \\ \square & \vdots & \square \\ a_{1p} & \square & a_{mp} \end{pmatrix} \quad (15)$$

また、データ行列 Y はデータを並べたものであるため、各データは次のような形で表現される。

$$\begin{pmatrix} | \\ \mathbf{y}_i \\ | \end{pmatrix} = a_{i1} \begin{pmatrix} | \\ \mathbf{u}_1 \\ | \end{pmatrix} + a_{i2} \begin{pmatrix} | \\ \mathbf{u}_2 \\ | \end{pmatrix} + \cdots + a_{ip} \begin{pmatrix} | \\ \mathbf{u}_p \\ | \end{pmatrix} \quad (16)$$

上式がモードの線形和で元のデータを表現した形であり、データ行列を構成する全てのデータはこのモードの線形和の形で完全に再現できる。つまり、共通のモードを使って線形和の係数 (POD 係数) のみを変化させることで、異なるデータを表現できることを意味する。なお、モードを全て使う必要はなく、データに対する貢献度 (特異値を使って判断可能) を考慮してモード数を削減することも可能である。

3.4 モードの利用

ここまでで、モードの抽出方法と元データを線形和で表現できることを説明したが、これをどのように使うかについて説明する。モードの情報の利用方法は多岐にわたるが、ここでは一つの例として数値解析の代理モデルに利用する方法について説明を進める。

データ行列を構成するデータが数値解析から得られた何かの物理量の空間分布だとする。問題を単純にするために、時間変化などは考えず、数値解析の入力パラメータや条件を変化させることで空間分布が変化する問題を想定する。なお、二つの入力パラメータ x_1 と x_2 を変化させて異なる空間分布が得られるものとする。複数ケースの数値解析を実施し、空間分布が得られた後に、データ行列が作成され、特異値分解によりモードと各ケースを表現する POD 係数が得られる。各ケースの違いは POD 係数の違いによってのみ表現されることになるため、POD 係数を入力パラメータと関連付けることで代理モデルを構築することが可能となる。つまり、図-8のように、異なる入力パラメータの組合せに対して異なる POD 係数が得られるため、入力パラメータを変数とする POD 係数の補間関数を構築 (式 (16) の係数 a を入力パラメータ x_1 と x_2 の関数として表現) すればよい。これができれば、任意の入力パラメータが与えられたとき、数値計算そのものを使わなく

ても、複数セットの数値計算の結果を参照する形で、結果を瞬時に出力できることを意味する。計算コストの高い数値解析を用いて数多くの計算が必要な場合には、このような代理モデルとしての使い方が効果的である。

入力パラメータと POD 係数の関連付けについては様々な方法 (例えば、先述のガウス過程回帰や後述のニューラルネットワークなども使用可能) が考えられるが、補間関数の性能が結果に大きく影響するため、フィッティング精度や事前に必要となるデータセットの数などに十分注意する必要がある。なお、代理モデルとして使用する際には、元のデータセットの範囲内 (ここでは変化させた入力パラメータの範囲内) においてはある程度の精度を保証しやすいが、外挿となる場合には精度が極端に低下する場合もあるため、その点は十分に留意すべきである。

4. ニューラルネットワーク

4.1 深層学習 (Deep Learning)

深層学習¹⁰⁾ は、動物の神経回路からヒントを得て提唱されたニューラルネットワークの計算により、大量のデータを学習し、その中に潜んでいる特徴などを自発的に獲得する手法である。

ニューラルネットワークに関する研究は 1940 年頃に始まり、50 年以上の歴史があるものの、うまくいかないと評価されていた手法であったが、改めて注目されるようになった理由は、近年の飛躍的な計算機能力の向上によって、以前と比べて大規模なニューラルネットワークの計算を実装することが可能となったためである。また、深層学習が学習の仕方などを、明示的にプログラムすることを必要としないことも注目を集める理由の一つである。

具体例として、画像から数字を認識させることが目的である場合を考える。このために深層学習を用いると、数字を認識するためのアルゴリズムを考案してプログラムする必要はなく、対象とする数字の写っている画像をニューラルネットで処理することで、数字を認識するためのルールを自発的に獲得する (プログラムにおいては、ニューラルネットワークにおける重みを自動で獲得する) ことができる。また、対象が数字

でなく、他の物体であっても基本的には同じであり、タスクの種類には大きく依存せず、汎用性があることができる。

4.2 ニューラルネットワーク

ニューラルネットは、図-9に示すようなユニットと呼ばれる素子からなっており、実数値の入出力を持っている。ユニットへの総入力は、式(17)のように重み付き和 U で表され、これを活性と呼ぶ。 w_i の値によって、各ユニットを結合させる際の強度は異なる。

$$U = \sum_i w_i x_i \tag{17}$$

この活性にバイアス b を加えて、活性化関数 f による変換を行ったものが、式(18)に示すユニットからの出力 z となる。

$$z = f(U + b) = f\left(\sum_i w_i x_i + b\right) \tag{18}$$

活性化関数としては、連続関数であるシグモイド関数がよく知られているが、最近では、正規化線形関数である ReLU(Rectified Linear Unit)やマックスアウトなどの関数が用いられることが多くなっている。

実際の処理においては、沢山の特徴量に重みをつけて足し合わせ、その結果を活性化関数によって非線形変換をすることで複雑な関数を作り出す。そして、さらにそれらを組み合わせることを繰り返し行うのが多層のニューラルネットワークで行われる処理である。

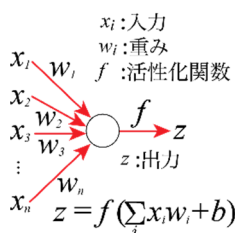


図-9 ユニットの構造

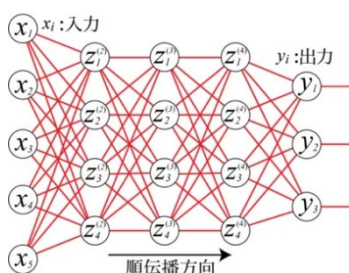


図-10 4層のニューラルネットワークの例

図-10に入力層と4層からなるニューラルネットワークの例を示す。深層学習では、ニューラルネットを多層化するため、深層という名称が使われている。ただし、多層化することによって、何故、良い学習機械が作れるのかを、現状では完全に説明しることができない。このため、深層学習から得られる結果を疑問視する意見があるのも事実である。今後は、深層化の優位性を示すことができる理論的な進展が期待されている。

4.3 勾配降下法による学習

深層学習、すなわち多層のニューラルネットワークにおける「学習」は、誤差関数 $E(\mathbf{w})$ の最小化問題を解くこととして定式化されるため、ネットワークパラメータ（重みとバイアス）のスカラ関数 $E(\mathbf{w})$ に最小値を与える $\mathbf{w} = \operatorname{argmin}_{\mathbf{w}} E(\mathbf{w})$ を求めることになる。ただし、誤差関数 $E(\mathbf{w})$ は一般に凸関数でなく、大域的な最小値を求めることは、ほぼ不可能なので、局所的な極小解 \mathbf{w} を求めることとなる。

また、 $E(\mathbf{w})$ の極小点は多数存在することが一般的で、偶然見つけた極小点が $E(\mathbf{w})$ の最小値を与えるとは限らない。極小解は、初期値 $\mathbf{w}^{(0)}$ を出発点に \mathbf{w} を繰返し更新する反復計算によって求められる。

非線形関数の最小化手法には様々なものがあるが、ここでは一般的に用いられることが多い簡単な方法として勾配降下法について説明する。勾配降下法における勾配はベクトルで、式 (19) のようになる。

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \equiv \left(\frac{\partial E(\mathbf{w})}{\partial w_1}, \dots, \frac{\partial E(\mathbf{w})}{\partial w_m} \right) \tag{19}$$

ここで m は \mathbf{w} の成分数である。勾配降下法では、現在の \mathbf{w} を負の勾配方向 $(-\nabla E)$ に動かすことを、何回も繰り返す。現在の重み $\mathbf{w}^{(t)}$ 、動かした後の重みを $\mathbf{w}^{(t+1)}$ とすると式(20)のように表される。

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \varepsilon \nabla E(\mathbf{w}^{(t)}) \tag{20}$$

ここで、 ε は1回の更新量の大きさを決めるハイパーパラメータで学習係数と呼ばれる。式(20)によって計算される $\mathbf{w}^{(t)}$ は、 ε が小さければ t の増加に伴って $E(\mathbf{w}^{(t)})$ が減少するので、何度も更新を繰り返すことによって極小点に到達することになる。ただし、学習係数 ε が小さすぎると \mathbf{w} の1回の更新量が小さすぎて、計算

コストが大きくなってしまいうので注意が必要である。また、 $E(\mathbf{w})$ の形状と ε の大きさによっては、 $E(\mathbf{w})$ の形状を捉える事ができず、増大することもあるため、 ε の決定は、学習を効率良く、かつ上手く行うための重要な項目の一つである。

4.4 訓練誤差, 汎化誤差, テスト誤差

学習の真の目的は、与えられた訓練データに対する誤差関数の値を最小化することではなく、未知のサンプル \mathbf{x} に対して正しい推定を行うことである。前者の訓練データに対する誤差のことを訓練誤差、後者の未知のデータに対する誤差（サンプルの母集団に対する誤差の期待値）のことを汎化誤差と呼んで区別する。

学習の目的は、汎化誤差を小さくすることであるが、汎化誤差は統計的な期待値であり、訓練誤差のように計算できない。このため、訓練データとは別のサンプル集合を未知のデータとして準備し、これに対して訓練誤差と同じ方法で計算される誤差を汎化誤差の目安とするが、これをテスト誤差と呼ぶ。

図-11 に学習によって訓練誤差とテスト誤差が変化する例を示す。一般的に、訓練誤差はパラメータ更新を繰り返すことで、ほぼ単調に減少する。一方で、テスト誤差は、最初のうちは訓練誤差と同じように減少するものの、途中から、訓練誤差とは異なる挙動となる。場合によっては、テスト誤差が訓練誤差と乖離してしまい、テスト誤差が増加してしまうこともある。このように訓練誤差とテスト誤差（汎化誤差）が乖離した状態は過学習と呼ばれる。

過学習を起こすことなく、テスト誤差が最小となっている状態が理想であるため、パラメータの更新に伴ってテスト誤差が増加する、もしくは一定となる場合は、それ以降の学習は過学習を引き起こすので、その時点で学習を終了させる。このように、途中段階で学習を終了させることを早期終了と呼び、ニューラルネットワーク学習のテクニックの一つである。

深層学習は、画像認識、自然言語処理、音声認識、異常検知などの様々な分野で活用されている。これらの目的に応じて、発展的なニューラルネットワーク（例えば、画像認識に用いられる畳み込みニューラルネットワーク（CNN）や、自然言語処理に用いられる回帰

結合型ニューラルネットワーク（RNN）など）に加えて、様々な手法が用いられるため、興味がある方は参考文献 10) を参照されたい。

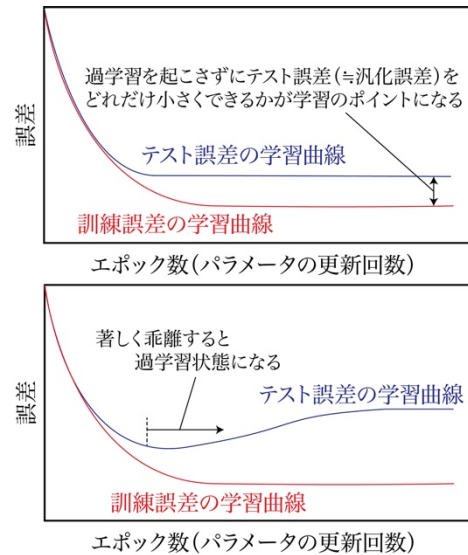


図-11 学習曲線のイメージ

5. まとめ

本稿「第4回 基礎編(その3)～予測モデルをつくる～」では、パラメータベクトルの推定問題や観測量ベクトルの将来予測を簡単にする予測モデル $f(\cdot)$ の作り方に焦点をあて、ベイズや機械学習導入のコツを例と共に概説した。近年多様な予測問題に適用されているガウス過程回帰と固有直交分解について述べ、その上で、最後に汎用的なツールとして多様な問題に適用できるニューラルネットワークの基礎を簡単に説明した。紙面の都合上、手法を網羅的に紹介できていないものの、予測モデルの作り方のコツを分類整理した本稿が読者の理解の一助となれば幸甚である。

表記法

- \mathbf{a} : パラメータベクトル (例題)
- a_{ij} : POD 係数
- b : ニューラルネットワークのバイアス
- c : 減衰定数 (例題)
- $E(\cdot)$: 誤差関数
- $f(\cdot)$: 観測量ベクトル \mathbf{y} の生成メカニズム

f : $f(\cdot)$ のベクトル表示
 κ, \mathbf{K} : カーネル関数
 k : ばね定数 (例題)
 \mathbf{k}^* : \mathbf{x}^* と \mathbf{x} の類似度を表すカーネル関数値のベクトル
 k^{**} : \mathbf{x}^* の自分自身の類似度
 m : 質点の質量 (例題)
 N : データ数
 p : ゼロではない特異値の数
 $\mathbf{u}_1, \mathbf{u}_2$: 主成分軸の基底ベクトル (特徴モード)
 U : ニューラルネットワークの重み付き和
 \mathbf{U} : 共分散行列の固有ベクトルを行方向に並べた行列
 w, \mathbf{w} : 重み
 x, \mathbf{x} : パラメータ, 入力値
 y, \mathbf{y} : 観測量, 観測値, データ行列
 $\varepsilon, \boldsymbol{\varepsilon}$: 誤差
 $\bar{\varepsilon}$: ニューラルネットワークの学習係数
 $\boldsymbol{\phi}, \Phi$: 特徴ベクトル
 $\boldsymbol{\Sigma}$: 分散共分散行列
 θ_1, θ_2 : カーネル関数の形を決めるパラメータ

- 2) Zhang, W., Li, H., Li, Y., Liu, H., Chen, Y. and Ding, X. : Application of deep learning algorithms in geotechnical engineering: a short critical review. *Artificial Intelligence Review*, Vol. 54, p. 5633-5673, 2021.
- 3) Baghbani, A., Choudhury, T., Costa, S. and Reiner, J. : Application of artificial intelligence in geotechnical engineering: A state-of-the-art review. *Earth-Science Reviews*, Vol. 228, p. 103991, 2022.
- 4) Estimation of trend and random components of conditional random field using Gaussian process regression, Yoshida, I., Tomizawa, Y., Otake, Y. : *Computers and Geotechnics*, Vol. 136, p. 104179, 2021.
- 5) Otake, Y., Shigeno, K., Higo, Y. and Muramatsu, S. : Practical dynamic reliability analysis with spatiotemporal features in geotechnical engineering, *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, Vol. 16, No. 4, 662-677, 2022
- 6) 持橋大地・大羽成征 : ガウス過程と機械学習 (機械学習プロフェッショナルシリーズ), 講談社, 2019.
- 7) Lumley, J. L. : The structure of inhomogeneous turbulent flows, *Atmospheric turbulence and wave propagation*, pp. 166-178, 1967.
- 8) Ha, D. M., Tkalich, P. and Chan, E. S. : Tsunami forecasting using proper orthogonal decomposition method, *Journal of Geophysical Research: Oceans*, Vol. 113, p. 6019, 2008.
- 9) 外里健太 : 広域災害ハザード評価のための数値解析とモード分解の融合, 東北大学博士論文, 2023.
- 10) Goodfellow, I., Bengio, Y. and Courville, A. : 深層学習, KADOKAWA, 2018.

参考文献

(原稿受理 2024.3.11)

- 1) Phoon, K-K. and Zhang, W. : Future of machine learning in geotechnics. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, Vol. 17 No.1, pp. 7-22, 2022.