データ駆動時系列モデリングによる微分方程式の導出 *

一橋大学商学部商学科
堤夏輝 (Natsuki Tsutsumi)

概要

適当なカオス的時系列データから背後にある微分方程式を推定することは一般に困難である．我々は微分方程式の推定手法を開発し，ローレンツ方程式のアトラクタ上でふるまう軌道の一変数のみの時系列データから微分方程式を推定した．推定した微分方程式を評価するため，短時間の時間発展予測に加えて，長時間発展によって得られた変数の出現頻度分布の再現を確認した．流体運動のマクロ変数に関する時間発展方程式を流体の基礎方程式であるナヴィエストークス方程式から解析的に導出することは困難なことが知られているが，同様の推定手法を用いて流体のエネルギー変数の時系列のみからその時間発展を描写する微分方程式モデルを得た．

1 はじめに

1.1 従来の時系列予測手法

時系列データから将来の変動を予測することは多くの場面で有用である. そして, その手法について今まで多くの研究がなされてきた. 今回はその中でも, 決定論的なモデルから生成されるデータの将来予測を行う手法について提案を行う.

多くの時系列予測はある時点 t における状態から次の時点 $t+1$ における状態を予測するものである. すなわち, 手元の時系列データ $\{\mathbf{X}_t\}$ を用いて,

$$\mathbf{X}_{t+1} \approx F(\mathbf{X}_t), \tag{1}$$

となるように関数 F を構成することが行われている. この F を構成する際に, ニューラルネットなどを用いたり, 入力を $\{\mathbf{X}_i | i \leq t\}$ としたりすることで, 多様な表現ができるようにしている [2].

このようにして作られた関数 F は別の力学系を構成していると考えられる. その力学的構造と元の力学的構造が似ていれば, 構成した関数 F は軌道だけでなく, 力学的構造もよく近似できているといえる. これはカオス性による短期予測の限界があったとしても, 本質的にモデリングには問題がないことを意味している. [4] では実際に力学的構造までも近似しうることを示している.

従来の式 (1) を考えることには 2 つの難点がある. 1 つ目の難点は時間刻みが変更できないことである. 多くの決定論的なモデルから生成されるデータはある微分方程式

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \tag{2}$$

* 本研究は [7] で発表予定である

の解軌道から時間刻み Δt ごとにサンプリングされたものと考えることができる. これを生成モデルと呼称する. 一般的には, 生成モデルの関数 f は時刻 t に依存しうるが, 今回は時刻に依存しない自励系を仮定する. このことから, 従来の手法によって推定された関数 F はデータの時間刻み Δt に依存する関数である. したがって, データの時間刻み Δt が変化した場合は関数 F を構成しなおす必要がある. このことは技術的に困難であるだけでなく, 本質的に同じものを異なるようにモデリングすることにあたる. また, 構成されたモデルから, より細かい時間刻みの予測ができない.

2 つ目の難点は構造の推定が難しいことである. 上述のように, 推定されたモデルの力学的構造が生成モデルの力学的構造と似ているかを確認することも重要である. 生成モデルは式 (2) で表される微分方程式なので, この式を偏微分したりすることで構造を計算できる. しかし, 推定されたモデルは 式 (1) で表される離散時間力学系であり, 元の微分方程式の力学的構造を推定することは困難である. したがって, 別に構造を推定する手法を考える必要がある.

以上の 2 点が従来の式 (1) の難点である. この 2 つを解決できる予測手法について考える.

1.2 微分方程式推定の概要

1.1 節で述べた難点の原因は, 連続時間で表される生成モデルを離散時間のモデルとしてモデリングしていることにある. したがって, 連続時間のままモデリングを行えば, そのような難点は解決すると考えられる. 具体的に今回は, 式 (1) の関数 F を推定するのではなく, 以下の式における関数 G を推定する:

$$\left.\frac{d\mathbf{X}}{dt}\right|_{\mathbf{X}=\mathbf{X}_t} \approx G(\mathbf{X}_t). \tag{3}$$

このようにすることで, 従来手法における難点は解決する.

次の手順をもって, 式 (3) の関数 G を推定する. 初めに時系列データ $\{\mathbf{X}_t\}$ を用いてそれぞれの時点における時間微分値を推定する. そのあと, その推定時間微分値を目標変数として, カーブフィッティングを行う. カーブフィッティングとは, G を異なる p 個の関数 $\phi_j(\mathbf{x})$ を用いて

$$G(\mathbf{x}) =< \boldsymbol{\beta}, \boldsymbol{\phi}(\mathbf{x}) >, \tag{4}$$

と表して, データからこの $\boldsymbol{\beta}(:= \{\beta_1, \cdots, \beta_p\}^{\mathrm{T}})$ を推定する手法のことを言う. ここで, $<, >$ は標準内積を表し, $\boldsymbol{\phi} := \{\phi_1, \phi_2, \cdots, \phi_p\}^{\mathrm{T}}$ とした. 例えば, [1, 8] ではこの手法を用いて, 単純なモデルを多項式で近似している. 詳しくは第 3 章に記している.

このようにすることで, 軌道だけでなく, 力学的構造も確かめることが可能なモデルをつくることができる.

1.3 レポートの構成

本レポートでは, 第 2 章で具体的な微分方程式を推定する手法を説明する. その後, 第 3 章にカーブフィッティングを多項式で行った場合の精度と限界について紹介する. そのあと, 第 4 章でその難点を解決するために動径関数を導入したものを紹介する. 最後に, 第 5 章で 流体のマクロダイナミクスのモデリングに手法を応用した例を説明する.

2　微分方程式推定手法

2.1　時間微分値の推定

初めに, 時系列データから特定の点での時間微分値の推定を行う手法について説明を行う. 手元の時系列データを $\{\mathbf{X}_t\}$ と表し, それが局所的に連続な関数 $\tilde{\mathbf{x}}(t)$ という関数で表せるとする. また, この関数が必要な階数微分可能であると仮定する. そこで, この関数を $2M$ 次の Taylor 展開をすると,

$$\tilde{\mathbf{x}}(\tilde{t}+h) \approx \sum_{n=0}^{2M} \left\{ \left. \frac{d^n \tilde{\mathbf{x}}}{dt^n} \right|_{t=\tilde{t}} \right\} \frac{h^n}{n!},$$

と近似できる. この関数をデータから求めることを考えると, $\{\mathbf{X}_t | t = \tilde{t} - M, \cdots, \tilde{t} + M\}$ を用いることで, 一意に求まる. そして, その時の h に関する 1 次項が時間微分値の推定値となる. 今回は $M = 3$ で微分値推定を行った. この場合, $\frac{d\mathbf{x}}{dt}|_{t=\tilde{t}}$ の推定値は以下のように計算される:

$$\left. \frac{d\mathbf{X}}{dt} \right|_{\mathbf{X}=\mathbf{X}_{\tilde{t}}}^{\mathrm{est}} = \frac{1}{60\Delta t}\{X_{\tilde{t}+3} - 9X_{\tilde{t}+2} + 45X_{\tilde{t}+1} - 45X_{\tilde{t}-1} + 9X_{\tilde{t}-2} - X_{\tilde{t}-3}\}. \tag{5}$$

2.2　最小二乗法

次にカーブフィッティングの手法を説明する. 以降の説明では, $y_t := \frac{d\mathbf{X}}{dt}|_{\mathbf{X}=\mathbf{X}_t}^{\mathrm{est}}, \phi_{(t,j)} := \phi_j(\mathbf{X}_t)$ とし, $\mathbf{y} := \{y_1, \cdots, y_N\}^{\mathrm{T}}, \Phi := \{\phi_{(t,j)}\}$ とする. ただし, N はデータ数である. 1.2 節で述べたように, カーブフィッティングは, 式 (4) の $\boldsymbol{\beta}$ を推定することである. その推定手法の一つに最小二乗法がある. 式 (4) によって G の値を近似するにあたり, 誤差が最小になるものが良いと考えられる. ここで, 時点 t での誤差を真の値と予測値の差の二乗と定義し, その平均値をモデルの誤差と考える. すると, $\boldsymbol{\beta}$ に関する誤差の関数 $(L(\boldsymbol{\beta}))$ は

$$L(\boldsymbol{\beta}) = \frac{1}{N}||\mathbf{y} - \Phi\boldsymbol{\beta}||_{l_2}^2, \tag{6}$$

となる. この関数を最小化するように $\boldsymbol{\beta}$ を決定する方法を最小二乗法という. 式 (6) は微分可能であるから, 最小化を行うパラメータ $\boldsymbol{\beta}^{\mathrm{LSE}}$ は以下のように表せる:

$$\boldsymbol{\beta}^{\mathrm{LSE}} = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}\mathbf{y}. \tag{7}$$

2.3　Ridge 回帰

データを生成する関数が実際に ϕ_j の線形結合で表される場合, 最小二乗法は良い手法である. 一方で, そうでない場合は過学習を起こしやすい. 過学習とは手元データにフィットしすぎることによって, 未学習データへの適用ができなくなってしまうことである. これを防ぐ為に, 正則化という手法が有用である. 正則化とは, 式 (6) に $\boldsymbol{\beta}$ が望ましい値であるほど小さい値をとる関数 $(R(\boldsymbol{\beta}))$ を足して最小化する方法である. 過学習を起こす場合, パラメータの絶対値が大きくなることが多いため, $R(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_{l_2}^2$ とすることで過学習を防ぐことができると考えら

れる．このような思想の元,

$$\hat{L}(\boldsymbol{\beta}) = \frac{1}{N}||\mathbf{y} - \Phi\boldsymbol{\beta}||_{l_2}^2 + \lambda||\boldsymbol{\beta}||_{l_2}^2, \tag{8}$$

を最小化する推定手法を Ridge 回帰という．この式において, λ は正則化の強さを決定するパラメータで, 正則化パラメータといわれる．この正則化の強さは $\mathbf{y}$ の分散が大きいほど, 相対的に弱くなる．なぜならば, 誤差が相対的に大きくなるからである．そこで, $\mathbf{y}$ を標準偏差で割って推定を行い, 複数の推定において共通した正則化パラメータを使用する．

式 (8) を最小化するパラメータ $\boldsymbol{\beta}^{\mathrm{Ridge}}$ は

$$\boldsymbol{\beta}^{\mathrm{Ridge}} = (\Phi^{\mathrm{T}}\Phi + \lambda N I)^{-1}\Phi^{\mathrm{T}}\mathbf{y}, \tag{9}$$

と表せる．この式において, I は p 次の単位行列を表す．

3　多項式モデリング

本節では, もっとも単純な推定手法として, $\phi_j(\mathbf{x})$ に多項式を用いた例を紹介する．その性能を評価するにあたり, Lorenz 微分方程式から生成されたデータを用いる．3.2 節までは [1, 8] を参考にしている．

3.1　Lorenz 微分方程式

Lorenz 微分方程式 [5] は熱対流を示す偏微分方程式を大胆に単純化することで現れる 3 次元の常微分方程式で, 以下で示される:

$$\begin{aligned} \frac{dx}{dt} &= p(y - x) \\ \frac{dy}{dt} &= rx - y - xz \\ \frac{dz}{dt} &= xy - bz. \end{aligned} \tag{10}$$

今回, データを生成するにあたり, $(p, r, b) = (10, 28, \frac{8}{3})$ とした．この設定はカオス性が生まれることで知られている．また, データの生成は 4 次の Runge-Kutta 法を用い, 時間刻み 0.005 で時間長 $5,000$ だけ行った．

3.2　多項式モデリングの精度評価

Lorenz 微分方程式の 3 次元データに対して, 高々 3 次の多項式回帰を実施した結果を紹介する．推定した結果の係数を表 1 に記した．それを見ると, 本来の微分方程式をよく近似できていることがわかる．実際に数値解軌道を計算すると, 本来の微分方程式の解軌道に近似することもわかる．

このことは, 生成モデルが低次の多項式で表せる場合に, それ以上の次数で回帰をかければよくモデルを近似できることを示している．しかし, 生成モデルが低次の多項式で表せる場合や表されたとしてもそのすべての変数を観測できる場合は多くない．そのような場合の評価を次節で行う．

回帰次数	X_1	X_2	X_3	X_1X_2	X_3X_1
第 1 変数	-10.000	10.000	0.000	0.000	0.000
第 2 変数	28.000	-1.000	0.000	0.000	-1.000
第 3 変数	0.000	0.000	-2.667	1.000	0.000

表 1　**Lorenz 微分方程式のモデリング結果.** 高々 3 次の多項式回帰を最小二乗法で行ったときの係数を示している. 推定された 3 つの回帰式すべてにおいて, 係数が絶対値が 10^{-4} より小さいものは省略した.

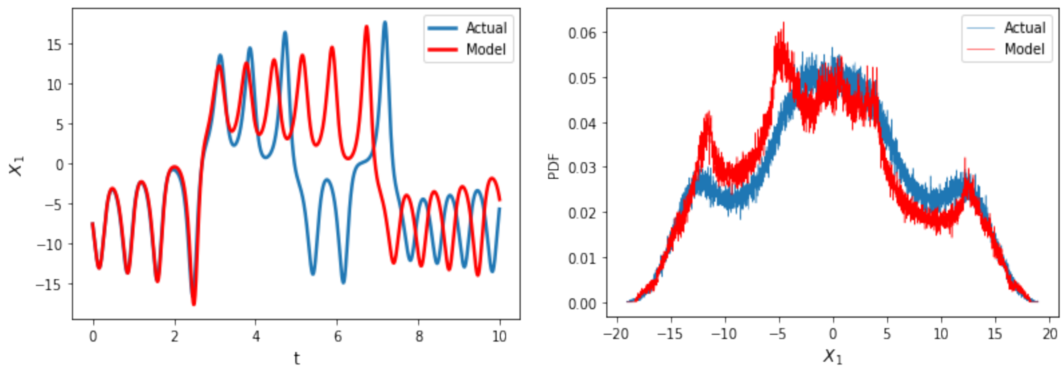

図 1　**時間遅れ系に対する多項式モデリングの結果.** 数値解を時間刻み 0.005 の 4 次 Runge-Kutta 法により求めた. 左図は時間長 10 の X_1 変数の軌道. 短期の予測はできていることがわかる. 右図は時間長 $5,000$ の軌道を求めたときの X_1 変数の頻度グラフ. 頻度グラフは本来のものを近似できていないことがわかる.

3.3　部分観測と時間遅れの導入

現実では, 生成モデルが低次の多項式で記述されるとは限らず, また一部の変数しか観測できない場合も多い. そのような場合に適用できる手法を考える. その例として, Lorenz 微分方程式のデータのうち, 第 1 変数しか手元にない場合を考える.

入手可能な変数が限られる場合, 埋め込みという操作が生成モデルを再現するのに有用であることが知られている [9]. 埋め込みとは, 1 次元の変数 $x(t)$ に対して, その時間遅れを第 2 変数や第 3 変数として追加する方法である. 例えば, 2 次元分を追加する場合は, 観測された系を疑似的に $(x(t), x(t-\tau), x(t-2\tau))$ とする. 適切に追加する次元と時間遅れの幅 τ を決定することができれば, この手法によって, 元のモデルの性質などを近似できることが知られている. この次元や時間遅れ幅の決定方法について様々な提案がされているが, 確立された方法はない. そこで, 今回は Lorenz 微分方程式の第 1 変数データを扱うにあたり, 時間遅れ幅 0.13 で 2 次元分を追加した.

このモデルに対して, 多項式モデリングを適用した結果を記す. 今回, 高々 8 次の多項式回帰を Ridge 推定で行った. その方程式の短期軌道を図 1 の左図に示した. それを見ると, 短期軌道の予測はある程度うまくいっていることがわかる. 一方で, 統計的特性はうまく近似できていないことが, 図 1 の右図によりわかる.

これらのことから, 多項式で表されるかわからないモデルを低次の多項式でモデリングすると, 短期の軌道は予測しえても, 長期の統計的特性まで近似できるモデルを作ることは簡単ではないと考えられる. そこで, それを解決する手法を次章で導入する.

4　動径関数

Lorenz 微分方程式の第 1 変数とその時間遅れを変数に持つ系の微分方程式を動径関数を用いて推定する.

4.1　動径関数の導入

3.3 節において多項式モデリングが困難であった理由の一つに, 局所的な構造をとらえられなかったことが考えられる. そこで, 局所的な構造を表現できる関数を ϕ_j に用いることを考える.

局所的な構造を表現する方法の 1 つに動径関数がある [3]. 動径関数とは, 入力 $\mathbf{x}$ に対して, あらかじめ決まっている点 $\mathbf{c}$ と $\mathbf{x}$ との距離のみに依存する関数のことをいう. そして, その関数を距離に対して単調減少な有界関数にすることで, 局所的に影響力を持つ関数を構成することができる. 今回はガウス型動径関数といわれる以下の関数 $g_k(\mathbf{x})$ を用いる:

$$g_k(\mathbf{x}) = \exp\left\{\frac{-||\mathbf{x}-\mathbf{c}_k||_{l_2}^2}{h^2}\right\}. \tag{11}$$

ここで, $\mathbf{c}_k$ は k 番目の中心点の座標, h は関数の広がりを表現するパラメータである.

今回, このガウス型動径関数を用いるにあたって, データの存在するすべての領域で 4 つの動径関数に覆われるように中心点 $\{\mathbf{c}_k\}$ を格子状に配置し, h を設定した. 動径関数は $1,806 (=: M)$ 個用い, その他に一次項も $\phi_j(\mathbf{x})$ に用いた. つまり, 今回は

$$\left.\frac{d\mathbf{X}}{dt}\right|_{\mathbf{X}=\mathbf{X}_i}^{\text{est}} \approx \beta_0 + \sum_{j=1,\cdots,D} \beta_j X_{(i,j)} + \sum_{k=1,\cdots,M} \beta_{D+k}\ g_k(\mathbf{X}_i), \tag{12}$$

として, $\boldsymbol{\beta}$ を推定した. ここで, $\mathbf{X}_{(i,j)}$ はベクトル $\mathbf{X}_i$ の第 j 変数を表す. また, 動径関数は過学習しやすいため, 推定手法には Ridge 回帰 (式 (9)) を用いた.

4.2　短期軌道の評価

動径関数を用いて推定された微分方程式の短期軌道を評価する. 図 2 に短期の解軌道を示している. 多項式モデリングの場合よりも短期予測の精度がよく, かつ時間遅れの構造も保っていることがわかる.

4.3　長期軌道の評価

動径関数を用いて推定された微分方程式の長期軌道を評価する. 図 3 に $10,000$ 時間の解軌道を 3 次元プロットした図を示した. 2 つのアトラクターがよく似ていることがわかる. また, 第 1 変数の頻度グラフを図 4 に示している. 多項式モデリングの場合は頻度グラフを近似できなかったが, 動径関数を用いることで頻度グラフも近似されている.

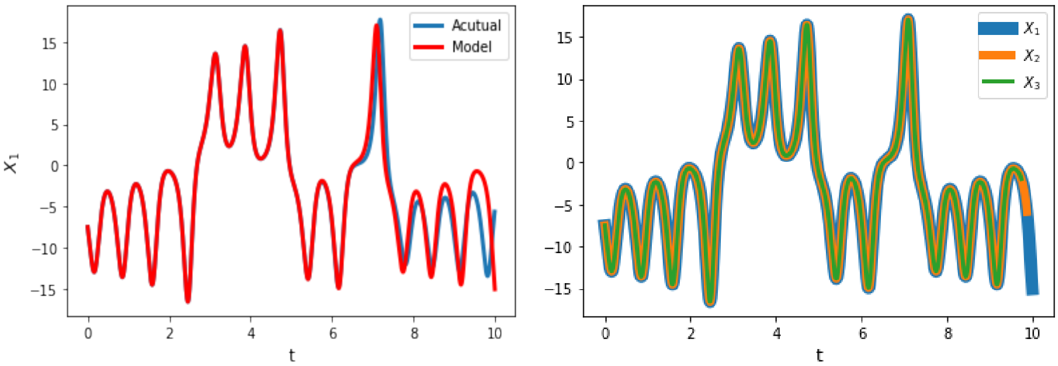

図 2　**動径関数を用いた場合の短期軌道．**左図は Lorenz 微分方程式の第 1 変数と今回推定された軌道の第 1 変数．予測がずれている部分はカオス性に由来するものと考えられる．右図は推定された微分方程式の軌道を時間遅れ分シフトさせた図．時間遅れ構造が推定された微分方程式にも維持されていることがわかる．

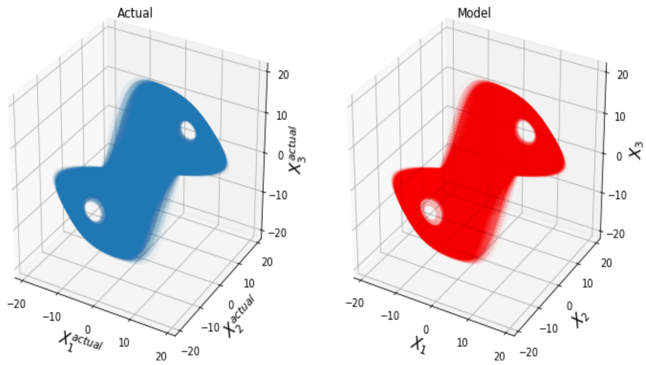

図 3　**動径関数を用いた場合のアトラクター．**時間刻み $\Delta t = 0.005$ で $T = 10,000$ だけ数値解を計算した結果．推定された微分方程式のものと本来のものが十分似ていることがわかる．

4.4　不動点

不変集合の一つである不動点についての比較を行う．不動点とは, 時間微分が 0 となる点のことで, その点が初期値であると動かない点である．

生成モデルである Lorenz 微分方程式 (式 (10)) から不動点をもとめると, 今回のパラメータ設定と時間遅れ系では $(0, 0, 0)$,$(8.485, 8.485, 8.485)$, $(-8.485, -8.485, -8.485)$ の 3 つが不動点となる．今回推定したモデルの不動点をオイラー法で求めると, 5 つの不動点が求まる．その座標とヤコビ行列の固有値を表 2 に示した．ヤコビ行列の固有値は不動点の性質を決定する要素である．本来の不動点の座標は良い精度で近似できていることわかる．一方で, 本来のモデルにはない不動点についてはアトラクターの外にあることが調べるとわかる．この 2 つの不動点をゴースト不動点と呼称する．

2 つのゴースト不動点はモデリング可能な境界を定めている．この不動点の形成する安定多様体を境界にアトラクター側に初期値があるとアトラクター内に吸引される．その一方で, ア

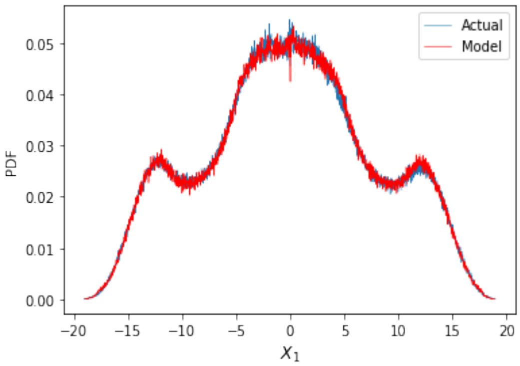

図 4　**動径関数を用いた場合の X_1 の頻度グラフ．**図 3 のデータ X_1 に関する頻度グラフ．本来の頻度グラフと似ていることがわかる．

	L	R	O	GL	GR
X_1^*	-8.48	8.48	0.00	-1.29	1.32
X_2^*	-8.47	8.47	0.00	-1.25	1.29
X_3^*	-8.48	8.48	0.00	-1.37	1.41
Λ_1^*	$0.10+10.17i$	$0.10+10.17i$	10.56	11.19	11.34
Λ_2^*	$0.10-10.17i$	$0.10-10.17i$	2.83	-5.12	-4.78
Λ_3^*	-10.60	-11.22	-15.76	-10.12	-10.62

表 2　**不動点の座標とヤコビ行列の固有値．**不動点の座標を (X_1^*, X_2^*, X_3^*) と表し，その点でのヤコビ行列の 3 つの固有値を $\Lambda_1^*, \Lambda_2^*, \Lambda_3^*$ としている．本来の不動点である L, R, O はよく近似できていることがわかる．また，本来の時系列データにないゴースト不動点 GL, GR が存在する．これらはアトラクターの外にあることが調べるとわかる．

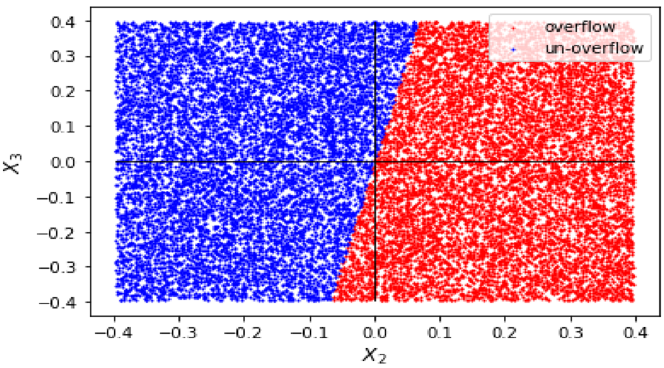

図 5　**ゴースト不動点の形成する境界面．**ゴースト不動点 GR の X_2, X_3 に乱数を足した場合の解軌道が発散するかを確かめた．10 時間以内にアトラクターを含む立方体から出た場合を発散とみなした．発散する場合は赤色，発散しない場合は青色でプロットしている．原点が GR にあたり，それを通る曲線が発散するかの境界面を形成していることがわかる．

トラクターのない側にあると発散すると考えられる．実際に，ゴースト不動点の周辺を初期値にとり数値軌道を求めた結果を図 5 に示した．それを見ると，中心のゴースト不動点を通る線で発散する領域としない領域が分割されていることがわかる．

4.5　まとめ

以上で見てきたように, 動径関数を導入することで短期軌道だけでなく, 長期の軌道の統計的性質や不動点といった性質も近似できる. また, その他にも Poincaré 写像や Lorenz 写像といった性質も元のモデルのものと似ていることが確認することができる. 動径関数を導入することで, 多項式モデリングでとらえられなかった局所的な構造がモデリングできるようになり, 様々な性質まで近似できるようになったと考えられる.

5　流体マクロダイナミクスのモデリング

第 4 章の手法を用いて, 流体力学のエネルギーに関連するマクロ変数 $E(3,t)$ の時系列 [6] を予測するモデルを構築した. 具体的に, 時間窓 5 の移動平均をとったデータに対して, 時間遅れ 2.0 で 4 次元分追加して微分方程式の推定を行った. 流体の基礎方程式からマクロ変数で閉じた方程式は解析的に得られないことが知られている. 一方で, 今回推定された微分方程式はマクロ変数の中で閉じたものとなっている. 図 6 に推定された微分方程式の数値解軌道を示した. ある程度の時間, もとの時系列を近似できることが確認できる.

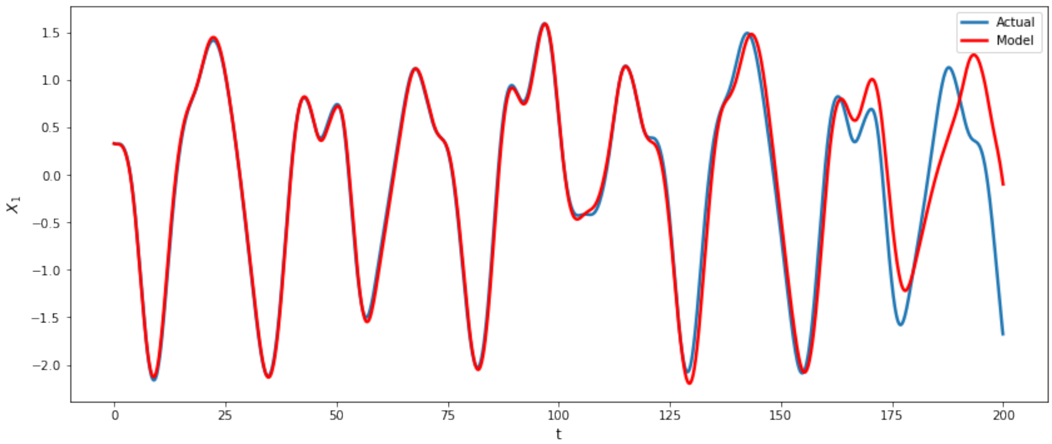

図 6　$X_1 = E(3,t)$ **の推定モデルの数値解軌道.** 時間刻み 0.005 の 4 次 Runge-Kutta 法で解軌道を計算した.

6　まとめ

今回, 生成モデルが未知のデータの微分方程式を推定する手法の提案を行った. データの時間微分値を Taylor 展開を用いて推定した後, それを目標変数にして動径関数に回帰することで近似的な微分方程式を作ることができる. この推定手法により求まった微分方程式は生成モデルの短期軌道だけでなく, 統計的な性質も近似しうることが分かった.

一方で, この手法は動径関数の配置を格子状に行う設定ゆえ, 大きな次元のデータおよび過度に複雑で多くの動径関数を必要とする場合などには適用できない. そのような場合にどのように行うべきかは今後検討するべき事項である.

謝辞

本研究は一橋大学大学院経営管理研究科齊木吉隆教授と東京海洋大学大学院学術研究院中井拳吾助教との共同研究である.

参考文献

[1] E. Baake, M. Baake, H. Bock, and K. Briggs. Fitting ordinary differential equations to chaotic data. *Physical Review A*, 45(8):5524, 1992.

[2] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[3] S. Kawano and S. Konishi. Nonlinear regression modeling via regularized gaussian basis function. *Bulletin of Informatics and Cybernetics*, 39:83–96, 2007.

[4] M. U. Kobayashi, K. Nakai, Y. Saiki, and N. Tsutsumi. Dynamical system analysis of a data-driven model constructed by reservoir computing. *Physical Review E*, 104:044215, 2021.

[5] E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20(2):130–141, 1963.

[6] K. Nakai and Y. Saiki. Machine-learning inference of fluid variables from data using reservoir computing. *Physical Review E*, 98(2):023111, 2018.

[7] N. Tsutsumi, K. Nakai, and Y. Saiki. Estimating differential equations from chaotic time-series. *in preparation*.

[8] W.-X. Wang, R. Yang, Y.-C. Lai, V. Kovanis, and C. Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106(15):154101, 2011.

[9] 池口徹. カオス時系列解析の基礎と応用. 産業図書, pages 33–58, 2000.