

## 動的モード分解による時系列データストリームの将来予測

千原 直己<sup>1,2,a)</sup> 松原 靖子<sup>1,b)</sup> 藤原 廉<sup>1,2,c)</sup> 櫻井 保志<sup>1,d)</sup>

受付日 2023年9月10日, 採録日 2024年1月5日

**概要:** 本論文では、動的モード分解を活用した大規模時系列データストリームの高速予測手法 MODECAST を提案する。MODECAST はセンサデータや、Web データなど、多種多様な時系列パターンにより構成される大規模時系列データストリームが与えられたとき、その中から潜在的なダイナミクスに基づいた重要な時系列パターンを発見することで将来予測を行う。より具体的には、このようなパターンを発見するために動的モード分解 (DMD) を活用する。提案手法は、(a) 大規模データストリームの中から重要なダイナミクスを発見し、リアルタイムかつ長期的な予測を可能とする。また、(b) 様々なデータに対して予測を行うことが可能であり、実用的である。さらに、提案手法は (c) データストリームの長さに依存せず、非常に高速である。実データを活用した実験により、提案手法が時系列データストリームの将来予測のための既存手法と比較して高精度であること、計算時間についてデータサイズに依存せず、より高速なリアルタイム予測を達成していることを明らかにした。

**キーワード:** 時系列予測, データストリーム処理, 動的モード分解

## Real-time Forecasting of Time-evolving Data Streams using Dynamic Mode Decomposition

NAOKI CHIHARA<sup>1,2,a)</sup> YASUKO MATSUBARA<sup>1,b)</sup> REN FUJIWARA<sup>1,2,c)</sup> YASUSHI SAKURAI<sup>1,d)</sup>

Received: September 10, 2023, Accepted: January 5, 2024

**Abstract:** Given a large, online stream of multiple co-evolving data sequences (e.g., sensor/web activities streams), which contains multiple distinct time-series patterns based on inherent dynamics, how do we capture important patterns and forecast future values? In this paper, we present MODECAST, an efficient and effective method for forecasting co-evolving data sequences. MODECAST exploits Dynamic Mode Decomposition (DMD) to capture time-series patterns based on inherent dynamics. Our proposed method has the following properties: (a) *Effective*: it captures important time-evolving patterns in data streams and enables real-time, long-range forecasting; (b) *General*: our model can be practically applied to various types of time-evolving data streams; (c) *Scalable*: our algorithm does not depend on the length of data streams and thus is applicable to very large sequences. Extensive experiments on a real dataset demonstrate that MODECAST makes long-range forecasts and consistently outperforms the best existing methods as regards accuracy, and the computational speed is sufficiently fast.

**Keywords:** time-series forecasting, stream processing, dynamic mode decomposition

## 1. はじめに

時系列データストリームは、IoT 関連の応用技術 [1], [2] や、web 活動 [3], [4], 疫病の感染 [5], ユーザの行動 [6] など、多くの事象から大量に生成されている。このようなデータの応用のための重要な技術の1つとして大量に発生し続ける時系列データストリームをリアルタイムに解析す

<sup>1</sup> 大阪大学産業科学研究所産業科学 AI センター  
SANKEN, Osaka University, Ibaraki, Osaka 567-0047,  
Japan

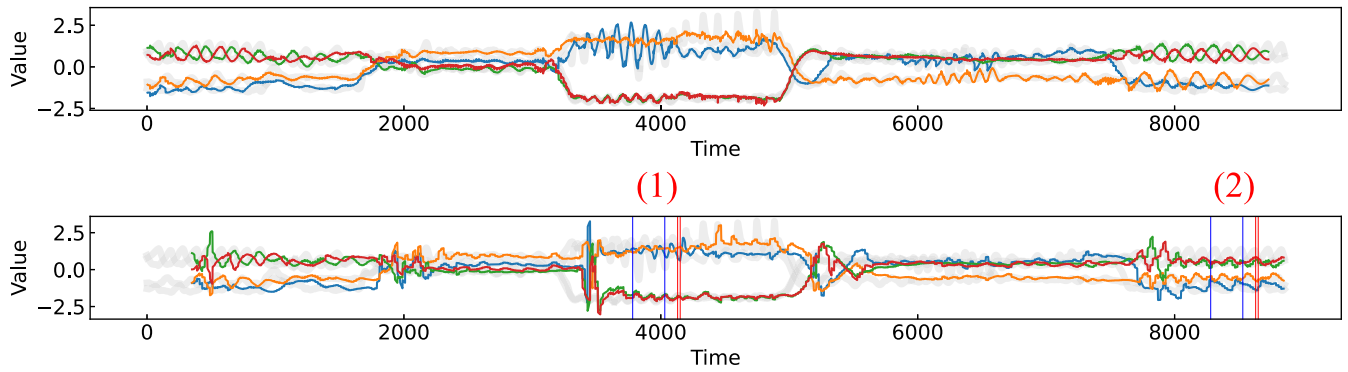
<sup>2</sup> 大阪大学大学院情報科学研究科  
IST, Osaka University, Suita, Osaka 565-0871, Japan

a) naoki88@sanken.osaka-u.ac.jp

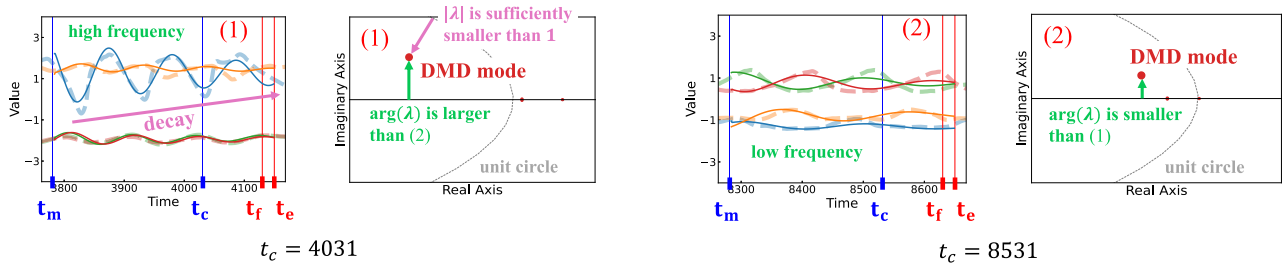
b) yasuko@sanken.osaka-u.ac.jp

c) r-fujiwr88@sanken.osaka-u.ac.jp

d) yasushi@sanken.osaka-u.ac.jp



(a) モーションイベントストリーム全体 (灰色の曲線) に対するフィッティング結果 (上) 及び予測結果 (下)



(b) 各時刻における  $l_s = 100$  ステップ先の予測結果のスナップショット及び動的モードに対応した固有値

**図 1** モーションイベントストリームに対する MODECAST による出力例：提案手法は継続的かつ自動的に  $l_s$  ステップ先の予測を行う。(a) 提案手法は時系列データストリームに対して良いフィッティングおよび予測を実現する。(b) 異なる 2 点の  $l_s$  ステップ先の予測の様子および予測に活用している動的モードに対応した固有値を示している。後者の動的モードに比べて前者の固有値の偏角の方が大きいことが、より高周波な成分を表現できていることを意味する

**Fig. 1** Forecasting results of MODECAST for motion sensor stream: It continuously and automatically forecasts  $l_s$ -steps-ahead-future values. (a) Our model fits and forecasts the original stream very well, (b) Snapshots of  $l_s$ -steps-ahead-future value prediction at two different time points and representative DMD eigenvalues for future value forecasting. A larger argument of the eigenvalue in the former DMD modes, compared to the latter, implies that MODECAST is capable of modeling higher-frequency components.

る技術があげられる。リアルタイム処理の中でも、時系列データストリームのリアルタイム予測は重要かつ挑戦的な課題である。

実際に、時系列データストリームには周期性やその増幅や減衰などといった様々な潜在的なダイナミクスが含まれており、高精度な将来予測を実現するためには、複数のダイナミクスから構成された重要な時系列パターンを効果的にとらえることは必要不可欠である。このために、我々は動的モード分解 [7], [8] を活用する。動的モード分解とは、時系列データを動的モードと呼ばれる要素に分解することにより、時間的なダイナミクスを抽出する数値解析手法である。具体的に、動的モードとは、発振周波数や固有値によって決定される減衰率/振動数と密接な関わりがあるため、動的モード分解では周期性やその増幅や減衰といったダイナミクスを効率的にとらえることができる。

また、時系列データストリームに含まれる様々な潜在的

ダイナミクスは刻一刻と変化する。たとえば、web の検索イベントデータには 1 年を通じた特徴的な時系列パターンの遷移 (e.g., ブラックフライデー、クリスマスなど) が見られることが多い。また、時期に応じて検索数には増減や減衰が見られる。そのため、より高精度に動的モード分解を活用した時系列データストリームのリアルタイム予測を実現するためには、潜在的なパターンの遷移もとらえる必要がある。本論文ではこのような潜在的時系列パターンのことを“レジーム”と呼ぶ。

本研究では、上記で述べたような要件を満たした時系列データストリームのリアルタイム予測手法 MODECAST を提案する。本論文では次の問題を扱う。

**問題：**  $d$  次元の観測データで構成される時系列データストリーム  $\mathbf{X} = \{\mathbf{x}(1), \dots, \mathbf{x}(t_c), \dots\}$  が与えられ、現時点を  $t_c$  としたとき、 $l_s$  ステップ先の  $\{\mathbf{v}(t_c + l_s), \dots, \mathbf{v}(t_c + l_s + l_p - 1)\}$  を予測する。

## 1.1 具体例

図 1 はモーションイベントストリーム (#2) *house-cleaning* における MODECAST の予測結果の様子を示している。このデータセットは 4 つのモーションキャプチャセンサ (左右の腕および足) から生成された “house cleaning” に関する複数のモーション (e.g., dragging a mop, wiping a window) から構成される。図 1 (a) は MODECAST のオリジナルデータストリーム (灰色の曲線) に対するフィッティング結果および予測結果 (それぞれの色付きの曲線) を示している。具体的には、各時刻  $t_c$  において  $\{\mathbf{x}(t_c + 100), \dots, \mathbf{x}(t_c + 119)\}$  の 20 点を予測し続けている (上記で定義した問題の  $l_s = 100$ ,  $l_p = 20$  のケース)。図中の青い線はそれぞれ  $\{t_m, t_c\}$ , 赤い線はそれぞれ  $\{t_f (= t_c + l_s), t_e (= t_c + l_s + l_p - 1)\}$  を示しており、 $t_c$  は現時刻を示す。この時系列データストリームには様々な明らかに異なる時系列パターンの変化が含まれているが、提案手法はその変化をよくとらえることができている。図 1 (b) は、2 つの異なる時刻における MODECAST の予測結果のスナップショットおよび予測のために使用されている複素平面上の動的モードの固有値 (点線は単位円) を示している。具体的には、スナップショットは  $t_c = 4031, 8531$  における出力の様子を示している。また、図中の固有値の絶対値はダイナミクスの減衰率を、偏角が振動数を表している。特に、減衰率に関しては、1 より小さい場合は減衰、1 より大きい場合は増幅に対応している。図を見て分かる通り、提案手法は時系列パターンの変化をとらえ、適切な予測を連続的かつ自動的に行っているといえる。また、2 枚のスナップショットを比較すると、周波数の観点から違いが明らかである。具体的には、 $t_c = 4031$  時点での時系列データは高周波であり、これを正確に表現できていることが確認できる。この観察を裏付ける形で、それぞれのスナップショットに対応する動的モードを比較したところ、固有値の偏角の大きさに顕著な違いがあることが分かる。さらに、前者のスナップショットに示される青色の曲線は減衰傾向にあるが、この点も固有値の絶対値を考慮することで、正確に表現できているといえる。

## 1.2 本論文の貢献

本論文では、大規模データストリームを効果的に表現する時系列モデル、および、その将来予測アルゴリズムである MODECAST を提案する。MODECAST は次の特長を持つ。

- 時系列データに含まれる特徴的なダイナミクスを動的モード分解により抽出することで、柔軟にレジームを発見し、最適かつリアルタイムな予測を実現する。
- 様々なデータに対して、実用的である。
- 計算コストはデータストリームの長さに依存することなく、高速に処理が可能である。

## 2. 関連研究

時系列のモデリングおよびその予測に関する研究は多岐にわたる。具体的な既存手法の例としては、自己回帰モデル (AR: autoregressive model), カルマンフィルタ (KF: Kalman filters) などがある [9]。加えて、このような古典的時系列解析の改良のための手法も多数提案されてきた [10], [11], [12], [13]。RegimeCast [14] は、生成され続ける観測データに対応した潜在的なパターンを非線形動的システムを適用することでリアルタイムに推定し、将来予測を行う。しかし、RegimeCast は周期性やその減衰率に基づいた時系列パターンを抽出することが困難である。また、深層学習を活用した時系列予測のための研究もさかんに行われている [15], [16], [17], [18], [19]。DeepAR [18] は、RNN ベースの確率的予測手法であり、複数系列のダイナミクスを考慮した予測が可能である。しかし、このような深層学習を基とした時系列予測手法は、モデルの学習の際に膨大な時間を要するため、リアルタイムにモデルを更新させ、現時刻の情報を取り入れながら予測を連続的に続けることは困難である。

動的モード分解に関する研究もさかんに行われている [20], [21], [22], [23], [24]。この手法は数値流体力学の分野で複雑な流体動力学データからコヒーレント構造を抽出するための手法として提案された [24]。また、動的モード分解には、非線形動力学系の線形であるが無限次元の表現を提供する Koopman 作用素 [25], [26], [27] のスペクトル解析と密接に関連しているという理論的背景がある。加えて、動的モード分解はデータ駆動型手法である。すなわち、潜在的なダイナミクスの抽出のために入力データ以外必要としない。そのため、たとえ、動的システムの物理モデルが不明だとしても、潜在的なダイナミクスを把握することができる。このような特性から動的モード分解は、非線形力学系から生成される時系列データの解析に有効であると考えられており、実際に、流体力学 [24]、動画処理 [28], [29]、脳波解析 [30]、疫学 [31] など、多くの分野で分析手法として広く採用されている。また、動的モード分解は時系列予測のためにも活用されている [32]。しかし、この手法は単次元データにしか対応していない。

## 3. 動的モード分解

本章では、動的モード分解について詳細に述べる。動的モード分解とは、時間発展するデータに含まれる潜在的な非線形ダイナミクスを抽出するための数値解析手法であり、出力結果は特定の条件下において Koopman 演算子 [25], [26], [27] によるモーダル分解と一致する。すなわち、対象のシステムの固有の減衰率/振動数を持つ特徴的な構造 (i.e., 動的モード) をとらえることが可能である。以下の式 (1) は動的モード分解の対象となる動的システム

を表している.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \quad (1)$$

上式における  $\mathbf{A}$  は動的システムの時間発展を表現するパラメータであり, これを解析することで重要な動的モードを抽出する. 入力時系列データ  $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^{d \times n}$  に対して以下のようなデータ行列を考える.

$$\mathbf{X}_0 = [\mathbf{x}_1 \quad \cdots \quad \mathbf{x}_{n-1}] \in \mathbb{R}^{d \times (n-1)}$$

$$\mathbf{X}_1 = [\mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n] \in \mathbb{R}^{d \times (n-1)}$$

式 (1) で表現される式が, 最も正しく表現される  $\mathbf{A} \in \mathbb{R}^{d \times d}$  は上記のデータ行列を用いて以下のように記述できる.

$$\begin{aligned} \mathbf{X}_1 \mathbf{X}_0^\dagger &= \arg \min_{\mathbf{A}} \sum_{k=1}^{n-1} \|\mathbf{x}_{k+1} - \mathbf{A}\mathbf{x}_k\|_2 \\ &= \arg \min_{\mathbf{A}} \|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\|_F \end{aligned}$$

ただし,  $\mathbf{X}_0^\dagger$  は  $\mathbf{X}_0$  の Moore-Penrose の擬似逆行列を,  $\|\cdot\|_F$  はフロベニウスノルムを示す. 以下では  $\mathbf{A}$  を効率的に解析するための動的モード分解によるアルゴリズムを述べる.

I.  $\mathbf{X}_0$  に対して特異値分解 (SVD) を計算する.

$$\mathbf{X}_0 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

ただし,  $\mathbf{U} \in \mathbb{R}^{m \times k}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ ,  $\mathbf{V} \in \mathbb{R}^{(n-1) \times k}$  であり,  $k$  は特異値の数を表している. 一般的に  $k$  は  $\mathbf{X}_0$  のランクで与えられるが, 一部削減を施すことにより観測される時系列データに通常含まれているノイズの影響を除去することが可能である.

II. ここで, より効率的な演算を行うために, 線形作用素  $\mathbf{A}$  を左特異値ベクトル  $\mathbf{U}$  が張る  $k$  ( $\leq \min(d, n)$ ) 次元部分空間への射影へを考える.

$$\tilde{\mathbf{A}} = \mathbf{U}^\top \mathbf{A} \mathbf{U} = \mathbf{U}^\top \mathbf{X}_1 \mathbf{V} \mathbf{\Sigma}^{-1}$$

III.  $\tilde{\mathbf{A}}$  の固有値分解を計算する. ここで, II. で使用した  $\mathbf{U}$  は直行行列であるため, ここで求めた固有値は  $\mathbf{A}$  の支配的な  $k$  個の固有値と一致する.

$$\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$$

IV. III. で求めた  $\mathbf{W}$  および  $\mathbf{\Lambda}$  を用いて  $\mathbf{A}$  の固有値分解を計算する. 具体的には, III. で述べたとおり,  $\mathbf{A}$  の固有値は  $\mathbf{\Lambda}$  で与えられ, 固有ベクトルは  $\mathbf{\Phi} = \mathbf{U}\mathbf{W}$  で与えられる\*1.

以上の手続きにより得られた  $\mathbf{\Phi}$  のことを動的モードという. また,  $\mathbf{\Lambda}$  は  $\Delta t$  (1 ステップ) 間の動的モードの減衰率/振動数を表現している. 具体的には, 一般的な線形動

\*1 このように  $\mathbf{\Phi}$  を与える動的モード分解を投影動的モード分解 (projected DMD) という.

表 1 主な記号と定義

Table 1 Symbols and definitions.

記号	定義
$t_c$	現在の時刻
$\mathbf{X}$	時系列データストリーム
$\mathbf{x}(t)$	時刻 $t$ における観測値
$\mathbf{X}^C$	カレントウィンドウ: $\mathbf{X}^C = \mathbf{X}[t_m : t_c]$
$\mathbf{X}^F$	予測ウィンドウ: $\mathbf{X}^F = \mathbf{X}[t_f : t_e]$
$h$	遅延座標系への射影のための遅延時間
$\mathbf{X}_{\text{aug}}$	遅延座標系で表現された時系列データストリーム
$\mathbf{\Lambda}$	固有値行列 $\mathbf{\Lambda} \in \mathbb{C}^{k \times k}$
$\mathbf{\Phi}$	動的モード $\mathbf{\Phi} \in \mathbb{C}^{dh \times k}$
$\mathbf{s}(t)$	時刻 $t$ における潜在値
$\mathbf{v}(t)$	時刻 $t$ における推定値
$\Theta$	全レジームのパラメータ集合: $\Theta = \{\theta_1, \dots, \theta_r\}$

的システムの解が  $e^{a+bi}$  ( $a, b \in \mathbb{R}$ ) の形式で表現できることに着目すると,  $i$  番目の動的モードの減衰率および振動数はこれに対応する固有値  $\lambda_i$  を用いて, それぞれ以下のような形式で表現できることが分かる.

$$\begin{aligned} \sigma_i &= \frac{\text{Re}(\log(\lambda_i))}{\Delta t} \\ \omega_i &= \frac{\text{Im}(\log(\lambda_i))}{\Delta t} \end{aligned}$$

ここで,  $\text{Re}$ ,  $\text{Im}$  はそれぞれ複素数の実部および虚部である.

## 4. ModeCast モデル

本章では, 本研究で扱う問題を定義し, その問題を解決するための MODECAST について述べる. まず, 本手法において必要な概念の定義について説明する. また, 表 1 に主な記号と定義を示す.

**定義 1 (カレントウィンドウ  $\mathbf{X}^C$ )**  $\mathbf{X}^C = \mathbf{X}[t_m : t_c]$ , すなわち時刻  $t_m$  から  $t_c$  までの区間で与えられる部分シーケンスを時系列予測に使用する最新のストリームデータとし, 長さ  $l_c = t_c - t_m$  のカレントウィンドウと呼ぶ.

**定義 2 (予測ウィンドウ  $\mathbf{X}^F$ )** 時刻  $t_c$  から  $l_s$  ステップ先の区間で与えられる  $\mathbf{X}^F$  を予測対象のデータ区間とし, 予測ウィンドウと呼ぶ. ここで,  $t_f = t_c + l_s$  を満たし, 予測ウィンドウの長さが  $l_p = t_e - t_f + 1$  のとき,  $l_p$  ステップごとに  $\mathbf{X}^F$  の値を予測する.

本論文では以下のような問題を取り扱う.

**問題 1 ( $l_s$  ステップ先予測)**  $t_c$  を最新の時刻とし, 長さが  $l_c$  のカレントウィンドウ  $\mathbf{X}^C = \mathbf{X}[t_m : t_c]$  が与えられる間,  $l_s$  ステップ先の長さ  $l_p$  の時系列データ  $\mathbf{X}^F = \mathbf{X}[t_f : t_e]$  を予測し続ける.

### 4.1 提案モデル

本研究の目的は, 時系列データストリームが与えられたとき, その中から重要かつ潜在的な動的モードに基づいた時系列パターン (レジーム) を発見し, リアルタイム予測

を実現することである。このような目的を達成するために MODECAST に対して以下の2つのアイデアを取り入れる。

(P1)：遅延座標への射影

(P2)：動的モードで表現されるレジーム

#### 4.1.1 遅延座標への射影 (P1)

本項では初めに、時系列データの状態空間について述べる。本手法では時間遅延座標系を用いた状態空間を利用する。時間遅延座標系とは、非線形方程式を推定するベースとなる技術であり、より長期的な潜在的ダイナミクスを把握するためにしばしば用いられている。具体的には、 $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$  に対して、以下のような行列  $\mathbf{X}_{\text{aug}}$  を考える。

$$\mathbf{X}_{\text{aug}} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{n-h+1} \\ \mathbf{x}_2 & \mathbf{x}_3 & \cdots & \mathbf{x}_{n-h+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_h & \mathbf{x}_{h+1} & \cdots & \mathbf{x}_n \end{bmatrix} \quad (2)$$

ただし、 $h$  は遅延時間を意味する。上記の行列の構造にあるとおり、 $\mathbf{X}_{\text{aug}}$  の状態ベクトルに過去の情報を付与できているため、位相情報を付与することが可能になり、これにより、より高精度にダイナミクスを表現することが可能になる。なお、Takens の埋め込み定理 [33] によると、このように遅延座標系で表された運動軌道は、時系列データを生み出した現実の位相空間上の運動軌道と同相な軌道となるため、3章で述べた動的モード分解のアルゴリズムに関して、 $\mathbf{X}$  を  $\mathbf{X}_{\text{aug}}$  に置き換えた結果得られる動的モードは適切なダイナミクスを表現する。

#### 4.1.2 動的モードで表現されるレジーム (P2)

次に、時系列パターンの表現方法について述べる。本研究では、潜在的ダイナミクスに基づいた時系列パターンを表現するために、複数の動的モードから構成されるレジームを定義する。このモデルにおけるレジーム  $\theta$  は次の2種類の状態によって構成されると考える。

- $\mathbf{s}(t) \in \mathbb{C}^k$ ：時刻  $t$  における  $k$  次元部分空間の潜在値
- $\mathbf{v}(t) \in \mathbb{R}^{dh}$ ：時刻  $t$  における  $dh$  次元の観測値

これらの要素を考慮した線形動的システムを次のように定義する。

$$\begin{aligned} \frac{d\mathbf{s}(t)}{dt} &= \mathbf{\Lambda}\mathbf{s}(t) \\ \mathbf{v}(t) &= \mathbf{\Phi}\mathbf{s}(t) \end{aligned} \quad (3)$$

ただし、初期条件は  $\mathbf{s}(0) = \mathbf{\Phi}^\dagger\mathbf{v}(0)$  で与えられる。また、 $\mathbf{\Lambda}, \mathbf{\Phi}$  はそれぞれ動的モード分解で得られる固有値および動的モードである。まとめると、以下を得る：

**定義3 (レジーム  $\theta$ )** 単一のレジームにおけるパラメータ集合  $\theta$  は右のとおりとする： $\theta = \{\mathbf{\Lambda}, \mathbf{\Phi}\}$ 。

さらに、我々は複数のレジーム  $\theta$  が時々刻々と変化する様子をとらえたい。そのために、時系列データストリームを複数のレジームで表現することを考える。そのために複

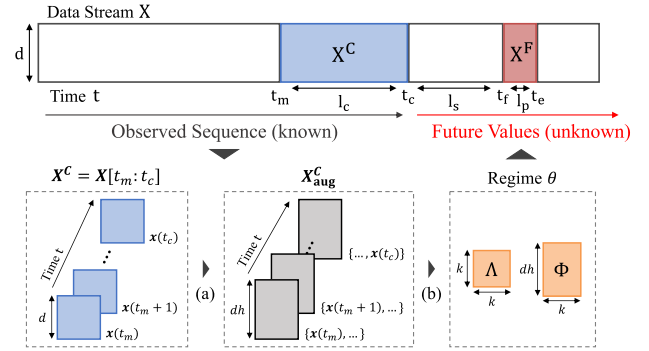


図2 MODECAST のモデル概要図：(a) カレントウィンドウ  $\mathbf{X}^C$  を遅延座標系へ射影し  $\mathbf{X}_{\text{aug}}^C$  を得る。その後、(b)  $\mathbf{X}_{\text{aug}}^C$  からレジーム  $\theta = \{\mathbf{\Lambda}, \mathbf{\Phi}\}$  を計算し、 $l_s$  ステップ先を予測する

Fig. 2 Overview of MODECAST: Given a current window  $\mathbf{X}^C$ , (a) It employs delay coordinate embeddings for obtaining  $\mathbf{X}_{\text{aug}}^C$ . (b) It estimates a regime  $\theta = \{\mathbf{\Lambda}, \mathbf{\Phi}\}$ , finally reports  $l_s$ -steps-ahead-future values  $\mathbf{V}^F$ , which are the estimated values of forecast window  $\mathbf{X}^F$ .

数のレジームで構成されたレジーム集合  $\Theta = \{\theta_1, \dots, \theta_r\}$  も同時に定義する。

提案モデルの全体図を図2に示す。提案モデルは遅延座標系および動的モードを活用することでカレントウィンドウ  $\mathbf{X}^C$  に最も適したレジーム  $\theta = \{\mathbf{\Lambda}, \mathbf{\Phi}\}$  を表現し、それを用いて将来予測を行う。

## 5. アルゴリズム

本章では、大規模時系列データストリームの将来予測アルゴリズムである MODECAST について述べる。

初めに、本アルゴリズムに必要な概念について定義を行う。

**定義4 (更新用パラメータ  $\omega$ )** 遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{\text{aug}}^C$  に対して、 $\mathbf{P} = (\mathbf{U}^\top \mathbf{X}_{\text{aug}_0}^C (\mathbf{U}^\top \mathbf{X}_{\text{aug}_0}^C)^\top)^{-1}$  とするとき、 $\omega = \{\mathbf{P}, \mathbf{U}\}$  を更新用パラメータと呼ぶ。ただし、 $\mathbf{U}$  は  $\mathbf{X}_{\text{aug}}^C$  の左特異ベクトルである。

**定義5 (全パラメータ集合  $M$ )** 全レジームの数を  $r$  としたとき、全レジームの集合  $\Theta = \{\theta_1, \dots, \theta_r\}$  に対応した更新用パラメータの集合を  $\Omega = \{\omega_1, \dots, \omega_r\}$  とすると、MODECAST のための全パラメータ集合  $M$  は  $\{\Theta, \Omega\}$  となる。

また、我々の最終目標としては、問題1の  $l_s$  ステップ先予測を逐次的、かつ高速に行うことである。ここで、現時刻  $t_c$  におけるレジームを  $\theta^c$ 、それに対応した更新用パラメータ集合を  $\omega^c$  とする。また、 $l_s$  ステップ先予測のために、時刻  $t_c$  における潜在値が必要であり、これを  $\mathbf{s}_{\text{out}}^c$  とする。以上より、これらをまとめて候補パラメータ  $\mathbf{C} = \{\theta^c, \omega^c, \mathbf{s}_{\text{out}}^c\}$  として保持する。

**Algorithm 1** MODECAST ( $\mathbf{x}(t_c), \mathbf{M}, \mathbf{C}$ )

---

```

1: Input: (a) New value  $\mathbf{x}(t_c)$  at time points  $t_c$ 
           (b) Parameter set  $\mathbf{M} = \{\Theta, \Omega\}$ 
           (c) Candidate  $\mathbf{C} = \{\theta^c, \omega^c, \mathbf{s}_{out}^c\}$ 
2: Output: (a) Updated parameter set  $\mathbf{M}'$ 
            (b) Updated candidate  $\mathbf{C}'$ 
            (c) Estimated variables  $\mathbf{V}^F$ 
3:  $\mathbf{X}^C \leftarrow \mathbf{X}[t_m : t_c]$  // Update current window  $\mathbf{X}^C$ 
4: /* Embed into time-delay coordinates */
5: Compute  $\mathbf{X}_{aug}^C$  according to Eq. (2)
6: /* Estimate optimal  $\theta^*$  */
7:  $\mathbf{M}', \mathbf{C}' \leftarrow \text{MODEESTIMATOR}(\mathbf{X}_{aug}^C, \mathbf{M}, \mathbf{C})$ 
8: /* Forecast  $l_s$ -step-ahead estimated value */
9:  $\mathbf{V}^F \leftarrow \text{MODEGENERATOR}(\mathbf{C}')$ 
10: return  $\{\mathbf{M}', \mathbf{C}', \mathbf{V}^F\}$ 
    
```

---

### 5.1 提案アルゴリズム

MODECAST は次のアルゴリズムで構成される (Algorithm 1).

- カレントウィンドウ  $\mathbf{X}^C$  から遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{aug}^C$  を生成する.
- MODEESTIMATOR: 遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{aug}^C$ , パラメータ集合  $\mathbf{M}$ , 候補パラメータ  $\mathbf{C}$  が与えられとき, 現在のデータに最も適したレジーム  $\theta^c$  を抽出する. また, パラメータ集合  $\mathbf{M}$  および候補パラメータ  $\mathbf{C}$  を更新する (Algorithm 2).
- MODEGENERATOR: 候補パラメータ  $\mathbf{C}$  が与えられたとき,  $l_s$  ステップ先の予測ウィンドウ  $\mathbf{V}^F$  を予測する.

### 5.2 ModeEstimator

遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{aug}^C$ , パラメータ集合  $\mathbf{M}$ , 候補パラメータ  $\mathbf{C}$  が与えられたときを考える. ここで, 損失関数を  $f(\mathbf{s}_0^c, \theta^c, \mathbf{X}_{aug}^C)$  と定義する. ただし,  $\mathbf{s}_0^c$  は時刻  $t_m$  における潜在値である. また,  $f(\cdot)$  は遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{aug}^C$  と, 遅延座標系で表現された推定ウィンドウ  $\mathbf{V}_{aug}^C$  のフィッティング精度を示す. 具体的には, 最小二乗誤差にて計算を行う (i.e.,  $f(\mathbf{s}_0^c, \theta^c, \mathbf{X}_{aug}^C) = \sum_{t=t_m}^{t_c} \|\mathbf{x}(t) - \mathbf{v}(t)\|^2$ ). 続いて, 具体的な  $\mathbf{V}_{aug}^C$  の計算方法について述べる. 式 (3) で与えられる線形動的システムの, レジーム  $\theta^c$  に対して初期値  $\mathbf{s}_0^c$  として  $\Phi^\dagger \mathbf{X}_{aug}^C[0]$  を与えれば計算自体は可能である. しかし, 初期値  $\mathbf{X}_{aug}^C[0]$  に過度なノイズが含まれていた場合, 適切な  $\mathbf{V}_{aug}^C$  を計算することができない. そこで, 我々は LM (Levenberg-Marquardt) アルゴリズム [34] を用いて  $\mathbf{s}_0^c$  を最適化することで, ノイズの影響の軽減により, より適切な  $\mathbf{V}_{aug}^C$  の計算を行う. ここで, 上記で述べたような初期値  $\mathbf{s}_0^c$  を最適化するための関数を  $g(\cdot)$  と定義する. また, この一連の過程を計算するための関数, すなわち,  $f(\cdot)$  と  $g(\cdot)$  を組み合わせた関数を  $\mathcal{F}(\cdot)$  と定義する.

**Algorithm 2** MODEESTIMATOR ( $\mathbf{X}_{aug}^C, \mathbf{M}, \mathbf{C}$ )

---

```

1: Input: (a) Current augmented window  $\mathbf{X}_{aug}^C$ 
           (b) Parameter set  $\mathbf{M}$ 
           (c) Candidate  $\mathbf{C}$ 
2: Output: (a) Updated parameter set  $\mathbf{M}'$ 
            (b) Updated candidate  $\mathbf{C}'$ 
3: if  $\mathcal{F}(\theta^c, \mathbf{X}_{aug}^C) > \rho$  then
4:   /* Find better regime in  $\Theta$  */
5:    $\{\mathbf{s}_0^c, \theta^c\} \leftarrow \arg \min_{\theta \in \Theta} \mathcal{F}(\theta, \mathbf{X}_{aug}^C)$ 
6:   if  $\mathcal{F}(\theta^c, \mathbf{X}_{aug}^C) > \rho$  then
7:     /* Create new regime */
8:     Compute  $\Lambda, \Phi$  according to Chapter 3
9:      $\theta^c \leftarrow \{\Lambda, \Phi\}; \Theta \leftarrow \Theta \cup \theta^c$ 
10:     $\omega^c \leftarrow \{P, U\}; \Omega \leftarrow \Omega \cup \omega^c$ 
11:     $\mathbf{s}_0^c \leftarrow g(\theta^c, \mathbf{X}_{aug}^C)$ 
12:   else
13:      $\{\theta^c, \omega^c\} \leftarrow \text{REGIMEUPDATE}(\theta^c, \omega^c, \mathbf{X}_{aug}^C)$ 
14:   end if
15: else
16:    $\mathbf{s}_0^c \leftarrow g(\theta^c, \mathbf{X}_{aug}^C)$ 
17:    $\{\theta^c, \omega^c\} \leftarrow \text{REGIMEUPDATE}(\theta^c, \omega^c, \mathbf{X}_{aug}^C)$ 
18: end if
19:  $\mathbf{M}' \leftarrow \{\Theta, \Omega\}$ 
20:  $\mathbf{C}' \leftarrow \{\theta^c, \omega^c, \mathbf{s}_{out}^c\}$ 
21: return  $\mathbf{M}', \mathbf{C}'$ 
    
```

---

MODEESTIMATOR は以下の手順に従う.

- (I)  $\mathcal{F}(\theta^c, \mathbf{X}_{aug}^C) > \rho$  となるとき, レジームセットからフィッティング精度を最小化するレジームを探索し,  $\theta^c$  を更新する.
- (II)  $\mathcal{F}(\theta^c, \mathbf{X}_{aug}^C) > \rho$  となるとき, 新たなレジームを生成し, レジームセット  $\Theta$  へ追加する.
- (III) (I), (II) の過程で得られた  $\mathbf{V}_{aug}^C$  の終端値  $\mathbf{s}_{out}^c$  を  $l_s$  ステップ先の予測のための初期値として, 候補パラメータ  $\mathbf{C}$  に保持する.
- (IV) 最適なレジームとして既存のレジームが採択された場合, REGIMEUPDATE に従って更新する.

ここで, 遅延座標系で表現されたカレントウィンドウ  $\mathbf{X}_{aug}^C$  に未知のパターンが出現したとき, 新たなレジーム  $\theta = \{\Lambda, \Phi\}$  を推定する必要がある. この推定にはそれぞれ 3 章で述べたアルゴリズムに従って推定する.

**オンラインレジーム更新.** 新たな観測データ  $\mathbf{x}(t_c)$  および 1 ステップ前のデータ  $\mathbf{x}(t_c - 1)$  から, 既存レジームの要素  $\Lambda, \Phi$  を更新用パラメータ集合  $\omega^c = \{P, U\}$  を活用して, 以下の流れで更新する.

- I'. 新たなデータおよびその 1 時刻前のデータを左特異ベクトル  $\mathbf{U}$  が張る  $k$  次元部分空間へ射影する

$$\tilde{\mathbf{x}}_1 \leftarrow \mathbf{U}^\top \mathbf{x}(t_c)$$

$$\tilde{\mathbf{x}}_0 \leftarrow \mathbf{U}^\top \mathbf{x}(t_c - 1)$$

- II'.  $\tilde{\mathbf{A}}$  を  $\mathbf{U}, \Lambda, \Phi$  から再構成し, 以下の式に従って更

新する.

$$\begin{aligned}\tilde{\mathbf{A}} &\leftarrow (\mathbf{U}^\top \Phi) \Lambda (\mathbf{U}^\top \Phi)^\dagger \\ \tilde{\mathbf{A}} &\leftarrow \tilde{\mathbf{A}} + \gamma (\tilde{\mathbf{x}}_1 - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_0) \tilde{\mathbf{x}}_0^\top \hat{\mathbf{P}}\end{aligned}$$

ただし,  $\gamma$ ,  $\hat{\mathbf{P}}$  とはそれぞれ以下のとおり.

$$\begin{aligned}\hat{\mathbf{P}} &= \frac{\mathbf{P}}{\delta} \\ \gamma &= \frac{1}{1 + \tilde{\mathbf{x}}_0^\top \hat{\mathbf{P}} \tilde{\mathbf{x}}_0}\end{aligned}$$

III'. 次の更新のために  $\mathbf{P}$  を更新する.

$$\mathbf{P} \leftarrow \frac{1}{\delta} (\hat{\mathbf{P}} - \gamma \hat{\mathbf{P}} \tilde{\mathbf{x}}_0 \tilde{\mathbf{x}}_0^\top \hat{\mathbf{P}})$$

IV'. 3章の III., IV. の手順どおりに更新後の  $\tilde{\mathbf{A}}$  から  $\Lambda$ ,  $\Phi$  を求める.

また, 現時刻  $t_c$  に近ければ近いほど, カレントウィンドウ  $\mathbf{X}^C$  のレジームを発見するために重要であるため, 更新の際, 新たな情報を付与するだけでなく, 過去の情報をモデルから忘却させる必要がある. そこで, 忘却係数  $\delta$  を導入する. 文献 [35] に従い,  $\delta$  は特別に明記しない限り  $\delta = 0.999$  とする.

### 5.3 ModeGenerator

次に, MODEESTIMATOR で推定された候補パラメータ  $\mathbf{C}$  が与えられたときを考える.

MODEGENERATOR は以下の手順で行われる.

- (I) 式 (3) に従って,  $\mathbf{s}_{out}^c$  を初期値として, 時刻  $t_c$  までの推定値をレジーム  $\theta^c$  を利用して遅延座標系で表現された推定ウィンドウ  $\mathbf{V}_{aug}^F$  を生成する.
- (II) 遅延座標系で表現された推定ウィンドウ  $\mathbf{V}_{aug}^F$  から推定ウィンドウ  $\mathbf{V}^F$  を復号する.

## 6. 評価実験

本論文では, MODECAST の有効性を検証するため, 実データを用いた実験を行った. 本章では以下の項目について検証する.

- Q1 データストリームの予測に対する提案手法の有効性
- Q2 リアルタイム予測に対する提案手法の精度の検証
- Q3 データストリームの予測に対する計算時間の検証

また, 実験は 512 GB のメモリ, Xeon Gold 6258R 2.7 GHz の 28 コア CPU および RTX A6000 48 GB GPU を搭載した Linux マシン上で実施した. 各データセットは平均値と分散で正規化 (z-normalization) して使用した.

**比較手法.** 有効性を検証するために, 以下の比較手法と比較を行った.

- RegimeCast [14]: 非線形動的システムに基づく将来予測アルゴリズム. 提案論文に従い, 潜在状態の数  $k = 4$ , モデルの階層数  $h = 2$ , モデル推定の閾値

表 2 データセット詳細

Table 2 Dataset description.

データセット	属性・クエリ	次元
センサデータ: モーションキャプチャにより取得		
(#1) <i>exercise</i>	左右の腕と足の動き	4
(#2) <i>house-cleaning</i>	"	4
(#3) <i>chicken-dance</i>	"	4
Web データ: GoogleTrends により取得		
(#4) <i>G-outdoor</i>	skiing, fishing, cycling	3
(#5) <i>G-sports</i>	tennis, American football, Major League Baseball (MLB)	3
(#6) <i>Beer</i>	Corona, Keystone, Coors, Modelo, SierraNevada	5
(#7) <i>OnlineTV</i>	Netflix, Hulu, YouTube, Amazon(P)	4
(#8) <i>Social media</i>	Tumblr, Facebook, LinkedIn, SoundCloud, Yelp	5
(#9) <i>Software</i>	HTML, Java, SQL, JavaScript, HTML5	5

$\epsilon = 0.5 \cdot \|\mathbf{X}^C\|$  とした.

- ARIMA: 古典的な時系列予測手法の 1 つ. 適切なパラメータ数は [14] に従い, AIC を用いて決定した.
- DeepAR [18]: 時系列予測のための最新のニューラルネットワーク手法. リカレントニューラルネットワーク (RNN) の層の数の候補は 1, 2, ユニット数の候補は 8, 16, 32, 64 とした. また, 最適化アルゴリズムとしては Adam [36] を使用し, 学習率を 0.01 で固定した. 最大エポック数 100 で, 適切な early stopping を利用し学習を行った結果, 最もフィッティング性能が良いモデルを採用した.

**データセット.** 実験に使用した 2 種類, 計 9 つの実データセットは次のとおりである. また, データセットの詳細を表 2 に示す.

- (#1) *exercise*, (#2) *house-cleaning*, (#3) *chicken-dance*: 様々な動きをする人間の様子をモーションキャプチャによって収集したセンサデータ.
- (#4) *G-outdoor*, (#5) *G-sports*, (#6) *Beer*, (#7) *OnlineTV*, (#8) *Social media*, (#9) *Software*: GoogleTrends から毎週収集された様々なクエリ (i.e., キーワード) の検索数によって構成されている.

### 6.1 Q1: 提案手法の有効性

本節では, 多次元時系列データストリームに対する MODECAST の予測性能を検証する. 実際のモーションイベントストリームにおける MODECAST の予測結果については 1 章の図 1 で説明したとおりであり, 提案手法はレジームの時間変化をとらえることによって, 高精度なリアルタイム将来予測を実現している. 特に, 図 1(a) のとおり, MODECAST は与えられた時系列データから適切なレジームをとらえ, それを予測に活かすことができている. また, 図 1(b) では, 各時刻における MODECAST のリアルタイム

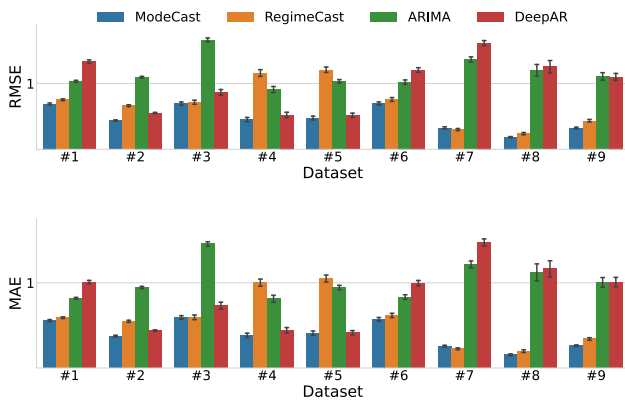


図 3 MODECAST の平均予測精度の比較：提案手法は既存手法と比べて高精度な予測が実現できている

Fig. 3 Average forecasting accuracy of MODECAST: our method is consistently superior to its competitors (lower is better).

ムに  $l_s = 100$  先の予測をしている様子およびその予測のために使用されている複素平面上の動的モードの固有値  $\Lambda$  が描画されている。この図は、寄与率の高い固有値が、時系列データストリームに含まれる周期性およびその増幅や減衰を適切に表現できていることを明らかにしている。

### 6.2 Q2：提案手法の精度

本節では、MODECAST の  $l_s$  ステップ先の予測精度を検証するため、比較手法である RegimeCast, ARIMA および DeepAR と精度比較を行った。長期のモーションイベントストリーム (#1, #2) においては  $l_s = 100$ , 短期のモーションイベントストリーム (#3) においては  $l_s = 30$ , web の検索イベントデータ (#4-#9) においては  $l_s = 13$  (i.e., 3 カ月先) と設定した。すべての手法において、パラメータチューニングのためにシーケンス全体の 3 分の 1 を訓練データとして与え、残りのデータに対して将来予測を行った。また、評価指標としては推定値の二乗平均誤差 (RMSE: root mean square error) および平均絶対誤差 (MAE: mean absolute error) の 2 種類を利用している。図 3 は、多次元時系列データストリームにおける MODECAST および比較手法の予測誤差の平均値を示している。図に示すとおり、提案手法は比較手法である RegimeCast, ARIMA および DeepAR と比較し、2つの評価指標のどちらにおいても高い予測精度を持つ。RegimeCast は固有値に基づいた潜在的なダイナミクスをとらえられていないため、予測精度が低下している。特に、web の検索イベントデータのうち、(#4, #5) のような固有の周期を持つと考えられるデータに対して予測精度が低下する。ARIMA は時系列データ間の線形関係を前提としたモデルであるため、複雑かつ非線形なデータに対応できず予測精度が低下する。また、DeepAR は逐次的なモデルの更新ができないため、レジームの遷移に対応できない。このため、予測精

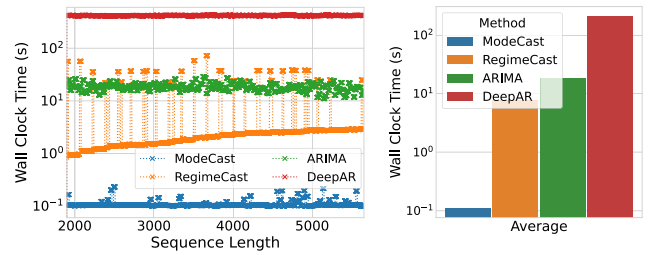


図 4 各時刻  $t_c$  における計算コスト (左) と平均値 (右)：これらの図の  $y$  軸は対数スケールで表示されている

Fig. 4 Scalability of MODECAST: (left) Wall clock time vs. data stream length  $t_c$ . (right) Average wall clock time. The vertical axis of these graphs is a logarithmic scale. MODECAST surpasses its competitors at any time.

度が低下しており、特に (#1) のような複数の明らかに異なるレジームを持つデータに対する予測精度が低下する。

### 6.3 Q3：提案手法の計算時間

続いて、提案手法の計算コストについて検証する。図 4 の左図は、(#1) *exercise* に対する各時刻  $t_c$  における計算コストを、提案手法の MODECAST と比較手法である RegimeCast, ARIMA および DeepAR と比較したものである。また、図 4 の右図は、データストリーム全体の計算時間の平均値を示している。これらの図の  $y$  軸は対数スケールで表示している。図 4 が示すとおり、MODECAST はデータストリームの長さ依存せずに、高速に動作することが可能である。特に、提案手法が高速である主な要因は、固有値分解のために部分空間への射影を行っている点である。実際に、固有値分解の計算コストは  $O(n^3)$  であるため、行列のサイズを小さくするフェーズは本手法の高速化に大きく貢献していると考えられる。一方、各比較手法は適切なレジームの学習のための処理に時間を要するため、MODECAST と比べて低速である。RegimeCast については新たなレジームを推定するフェーズに多くの時間を要する。実際、図 4 が示すとおり、RegimeCast の予測の大半が 1~5 秒程度で処理が完了しているのに対して、新たなレジームを推定しているであろう予測段階の処理時間が 100 秒弱に達している。これが全体の平均が約 10 秒となっている原因である。また、ARIMA および DeepAR は逐次的なモデルの更新ができないため、各時刻におけるストリーム処理に対する計算に一定の時間を要する。そのため、高速な時系列予測が実現できていない。以上より、提案手法は時系列データストリームのリアルタイム予測に適した性能を有するといえる。

## 7. むすび

本論文では、動的モード分解を活用した大規模時系列データストリームの高速予測手法として MODECAST を提案した。MODECAST は冒頭で確認した以下の優れている



特性をすべて達成している。

- 時系列データに含まれる特徴的なダイナミクスを動的モード分解により抽出することで、柔軟にレジームを発見し、最適かつリアルタイムな予測を実現する。
- 様々なデータに対して、実用的な予測を行うことが可能である。
- 計算コストはデータストリームの長さに依存することなく、高速に処理が可能である。

また、実データを用いた実験では、提案手法の性能を評価し、既存の時系列データストリームの将来予測のための既存手法と比較して予測精度、計算時間ともに改善していることを確認した。

謝辞 本研究の一部は JSPS 科研費, JP21H03446, 国立研究開発法人情報通信研究機構委託研究 (NICT) 委託研究 JPJ012368C03501, JSTAIP 加速課題 JPMJCR21U4, JST CREST JPMJCR23M3, の助成を受けたものです。

## 参考文献

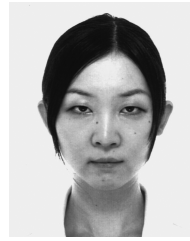
- [1] De Francisci Morales, G., Bifet, A., Khan, L., Gama, J. and Fan, W.: Iot big data stream mining, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.2119–2120 (2016).
- [2] Mahdavi, M.S., Rezvan, M., Barekatin, M., Adibi, P., Barnaghi, P. and Sheth, A.P.: Machine learning for Internet of Things data analysis: A survey, *Digital Communications and Networks*, Vol.4, No.3, pp.161–175 (2018).
- [3] Kawabata, K., Matsubara, Y., Honda, T. and Sakurai, Y.: Non-Linear Mining of Social Activities in Tensor Streams, *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2093–2102 (2020).
- [4] Nakamura, K., Matsubara, Y., Kawabata, K., Umeda, Y., Wada, Y. and Sakurai, Y.: Fast and Multi-aspect Mining of Complex Time-stamped Event Streams, *Proc. ACM Web Conference 2023*, pp.1638–1649 (2023).
- [5] Kimura, T., Matsubara, Y., Kawabata, K. and Sakurai, Y.: Fast Mining and Forecasting of Co-evolving Epidemiological Data Streams, *Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp.3157–3167 (2022).
- [6] Matsubara, Y., Sakurai, Y., Faloutsos, C., Iwata, T. and Yoshikawa, M.: Fast mining and forecasting of complex time-stamped events, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12–16, 2012*, Yang, Q., Agarwal, D. and Pei, J. (Eds.), pp.271–279, ACM (2012).
- [7] Tu, J.H.: *Dynamic mode decomposition: Theory and applications*, PhD Thesis, Princeton University (2013).
- [8] Kutz, J.N., Brunton, S.L., Brunton, B.W. and Proctor, J.L.: *Dynamic mode decomposition: Data-driven modeling of complex systems*, SIAM (2016).
- [9] Durbin, J. and Koopman, S.J.: *Time Series Analysis by State Space Methods*, OUP Catalogue, No.9780199641178, Oxford University Press (2012).
- [10] Li, L., Prakash, B.A. and Faloutsos, C.: Parsimonious linear fingerprinting for time series, *Proc. VLDB Endowment*, Vol.3, No.1-2, pp.385–396 (2010).
- [11] Shi, Q., Yin, J., Cai, J., Cichocki, A., Yokota, T., Chen, L., Yuan, M. and Zeng, J.: Block Hankel tensor ARIMA for multiple short time series forecasting, *Proc. AAAI Conference on Artificial Intelligence*, Vol.34, No.4, pp.5758–5766 (2020).
- [12] Papadimitriou, S., Brockwell, A. and Faloutsos, C.: Adaptive, hands-off stream mining, *Proc. 2003 VLDB Conference*, pp.560–571, Elsevier (2003).
- [13] De Livera, A.M., Hyndman, R.J. and Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing, *Journal of the American Statistical Association*, Vol.106, No.496, pp.1513–1527 (2011).
- [14] Matsubara, Y. and Sakurai, Y.: Regime shifts in streams: Real-time forecasting of co-evolving time sequences, *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1045–1054 (2016).
- [15] Nie, Y., Nguyen, N.H., Sinthong, P. and Kalagnanam, J.: A Time Series is Worth 64 Words: Long-term Forecasting with Transformers (2022).
- [16] Yue, Z., Wang, Y., Duan, J., Yang, T., Huang, C., Tong, Y. and Xu, B.: Ts2vec: Towards universal representation of time series, *Proc. AAAI Conference on Artificial Intelligence*, Vol.36, No.8, pp.8980–8987 (2022).
- [17] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H. and Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting, *Proc. AAAI Conference on Artificial Intelligence*, Vol.35, No.12, pp.11106–11115 (2021).
- [18] Salinas, D., Flunkert, V., Gasthaus, J. and Januschowski, T.: DeepAR: Probabilistic forecasting with autoregressive recurrent networks, *International Journal of Forecasting*, Vol.36, No.3, pp.1181–1191 (2020).
- [19] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J. and Long, M.: TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis, *International Conference on Learning Representations* (2023).
- [20] Kawahara, Y.: Dynamic mode decomposition with reproducing kernels for Koopman spectral analysis, *Advances in Neural Information Processing Systems*, Vol.29 (2016).
- [21] Takeishi, N., Kawahara, Y., Tabei, Y. and Yairi, T.: Bayesian dynamic mode decomposition, *IJCAI*, pp.2814–2821 (2017).
- [22] Erichson, N.B., Mathelin, L., Kutz, J.N. and Brunton, S.L.: Randomized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, Vol.18, No.4, pp.1867–1891 (2019).
- [23] Lusch, B., Kutz, J.N. and Brunton, S.L.: Deep learning for universal linear embeddings of nonlinear dynamics, *Nature communications*, Vol.9, No.1, p.4950 (2018).
- [24] Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics*, Vol.656, pp.5–28 (2010).
- [25] Koopman, B.O.: Hamiltonian systems and transformation in Hilbert space, *Proc. National Academy of Sciences*, Vol.17, No.5, pp.315–318 (1931).
- [26] Mezić, I.: Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dynamics*, Vol.41, pp.309–325 (2005).
- [27] Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P. and Henningson, D.S.: Spectral analysis of nonlinear flows,

- Journal of Fluid Mechanics*, Vol.641, pp.115-127 (2009).
- [28] Erichson, N.B. and Donovan, C.: Randomized low-rank dynamic mode decomposition for motion detection, *Computer Vision and Image Understanding*, Vol.146, pp.40-50 (2016).
- [29] Grosek, J. and Kutz, J.N.: Dynamic mode decomposition for real-time background/foreground separation in video, arXiv preprint arXiv:1404.7592 (2014).
- [30] Rohleff, J.: An Incremental Approach to Dynamic Mode Decomposition for Time-Varying Systems with Applications to a Model for Erythropoiesis, Master's thesis, Universität Konstanz, Konstanz (2020).
- [31] Proctor, J.L. and Eckhoff, P.A.: Discovering dynamic patterns from infectious disease data using dynamic mode decomposition, *International Health*, Vol.7, No.2, pp.139-145 (2015).
- [32] Tirunagari, S., Kouchaki, S., Poh, N., Bober, M. and Windridge, D.: Dynamic mode decomposition for univariate time series: Analysing trends and forecasting (2017).
- [33] Takens, F.: Detecting strange attractors in turbulence, *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*, pp.366-381, Springer (2006).
- [34] Moré, J.J.: The Levenberg-Marquardt algorithm: Implementation and theory, *Numerical Analysis: Proceedings of the Biennial Conference Held at Dundee, June 28-July 1, 1977*, pp.105-116, Springer (2006).
- [35] Zhang, H., Rowley, C.W., Deem, E.A. and Cattafesta, L.N.: Online dynamic mode decomposition for time-varying systems, *SIAM Journal on Applied Dynamical Systems*, Vol.18, No.3, pp.1586-1609 (2019).
- [36] Kingma, D.P. and Ba, J.: Adam: A Method for Stochastic Optimization, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Bengio, Y. and LeCun, Y. (Eds.) (2015).



千原 直己

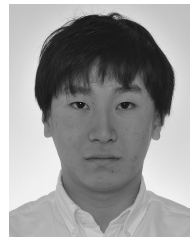
2023年大阪大学工学部電子情報工学科卒業。同年同大学院博士前期課程に進学。時系列データマイニングの研究に従事。



松原 靖子 (正会員)

2007年お茶の水女子大学理学部情報科学科卒業。2009年同大学院博士前期課程修了。2012年京都大学大学院情報学研究所社会情報学専攻博士後期課程修了。博士(情報学)。2012年NTTコミュニケーション科学基礎研

究所RA。2013年日本学術振興会特別研究員(PD)。2014年熊本大学大学院自然科学研究科助教。この間、カーネギーメロン大学客員研究員。2016年国立研究開発法人科学技術振興機構さきがけ研究者。2019年5月より大阪大学産業科学研究所准教授。2016年度日本データベース学会上林奨励賞、情報処理学会山下記念研究賞。2018年度IPSJ/ACM Award for Early Career Contributions to Global Research, 2020年度情報処理学会マイクロソフト情報学研究賞、電気通信普及財団第36回テレコムシステム技術賞、令和4年度科学技術分野の文部科学大臣表彰若手科学者賞等受賞。2018~2019年度日本データベース学会理事。大規模時系列データマイニングに関する研究に従事。ACM, IEEE, 電子情報通信学会, 日本データベース学会各会員。



藤原 廉

2021年大阪大学基礎工学部情報科学科卒業。2023年同大学院博士前期課程修了。同年同大学院博士後期課程に進学。時系列データマイニングの研究に従事。



櫻井 保志 (正会員)

1991年同志社大学工学部電気工学科卒業。1991年日本電信電話(株)入社。1999年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。博士(工学)。2004~2005年カーネギーメロン大学客員研究員。2013~

2019年熊本大学大学院自然科学研究科教授。2019年大阪大学産業科学研究所産業科学 AI センターセンター長・教授。2022年より日本学術振興会学術システム研究センター主任研究員。本会平成18年度長尾真記念特別賞, 平成16年度および平成19年度論文賞, 電子情報通信学会平成19年度論文賞, 日本データベース学会上林奨励賞, ACM KDD best paper awards (2008, 2010年) 等受賞。データマイニング, データストリーム処理, センサーデータ処理, Web情報解析技術の研究に従事。ACM, IEEE, 電子情報通信学会, 日本データベース学会各会員。

(担当編集委員 坂野 遼平)