

# 動的モード分解を活用した高速将来予測アルゴリズム

千原 直己<sup>†,††</sup> 松原 靖子<sup>†</sup> 藤原 廉<sup>†,††</sup> 櫻井 保志<sup>†</sup>

<sup>†</sup> 大阪大学産業科学研究所 〒 567-0047 大阪府茨木市

<sup>††</sup> 大阪大学大学院情報科学研究科 〒 565-0871 大阪府吹田市

E-mail: †{naoki88,yasuko,r-fujiwr88,yasushi}@sanken.osaka-u.ac.jp

**あらまし** 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 データ等、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時、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

## 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一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る技術が挙げられる。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捉えることができる。

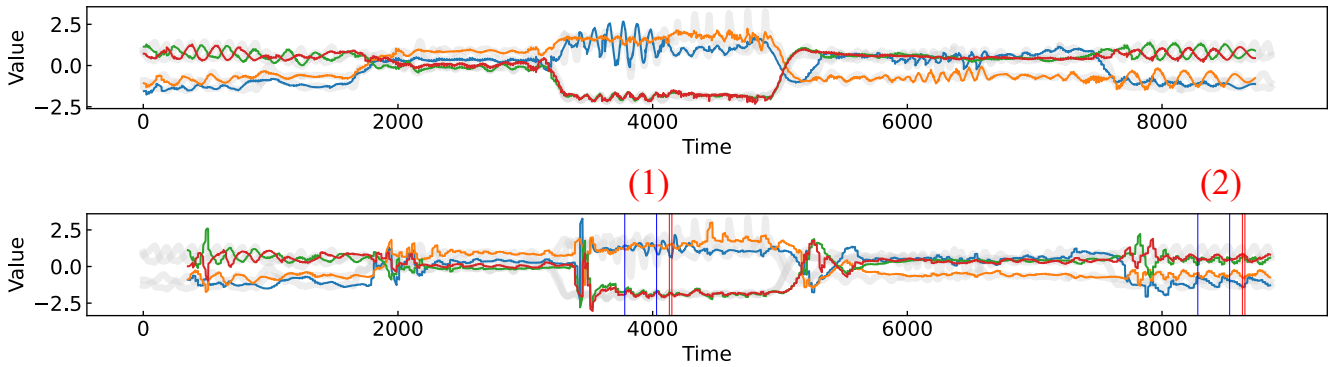
また、時系列データストリームに含まれる様々な潜在的ダイナミクスは刻一刻と変化する。例えば、web の検索イベントデータには 1 年を通じた特徴的な時系列パターンの遷移 (e.g., ブラックフライデー, クリスマスなど) が見られることが多い。また、時期に応じて検索数には増減や減衰が見られる。そのため、より高精度に動的モード分解を活用した時系列データストリームのリアルタイム予測を実現するためには、潜在的なパターンの遷移も捉える必要がある。本論文ではこのような潜在的時系列パターンのことを“レジーム”と呼ぶ。

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。

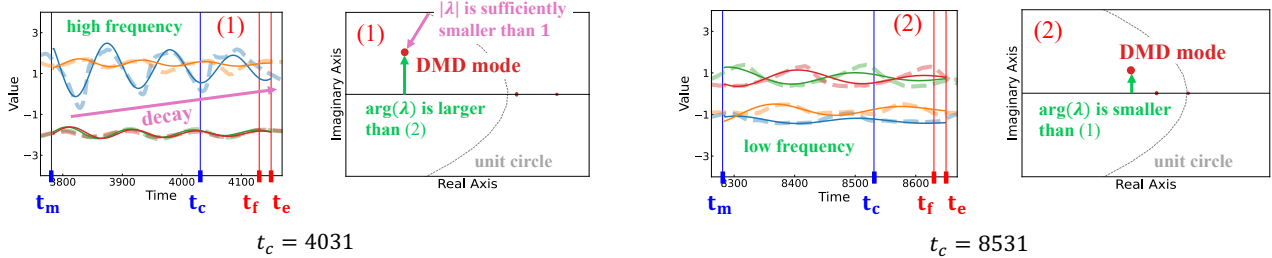
**問題:**  $d$  次元の観測データで構成される時系列データストリーム  $X = \{\mathbf{x}(1), \dots, \mathbf{x}(t_c), \dots\}$  が与えられ、現時刻を  $t_c$  とした時、 $l_s$  ステップ先の  $\{\mathbf{v}(t_c + l_s), \dots, \mathbf{v}(t_c + l_s + l_p - 1)\}$  を予測する。

### 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 (左右の腕及び足) 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及び予測結果 (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捉えることができている、それを適切に予測に活用できていると言える。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及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



(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及び予測を実現する。(b) 異なる 2 点の  $l_s$  ステップ先の予測の様子及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。

衰, 1 より大きい場合は増幅に対応している. 図を見てわかる通り, 提案手法は時系列パターンの変化を捉え, 適切な予測を連続的かつ自動的に行っていると見える. 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わかる. 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言える.

## 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, 及び, 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.

## 2 関連研究

時系列のモデリング及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 [10–13]. 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盛んに行われている [15–19]. 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行い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盛んに行われている [20–24]. 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 [25–27] のスペクトル

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

本節では、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 [25–27] によるモーダル分解と一致する。即ち、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) を捉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。

$$\mathbf{X}_0 = \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_{n-1} \end{bmatrix} \in \mathbb{R}^{d \times (n-1)}$$

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix} \in \mathbb{R}^{d \times (n-1)}$$

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。

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

$$= \arg \min_{\mathbf{A}} \|\mathbf{X}_1 - \mathbf{A}\mathbf{X}_0\|$$

但し、 $\mathbf{X}_0^\dagger$  は  $\mathbf{X}_0$  の Moore-Penrose の擬似逆行列を示す。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。

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

但し、 $\mathbf{U} \in \mathbb{R}^{m \times k}$ ,  $\mathbf{\Sigma} \in \mathbb{R}^{k \times k}$ ,  $\mathbf{V} \in \mathbb{R}^{(n-1) \times k}$  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

表 1 主な記号と定義

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: $\Theta = \{\theta_1, \dots, \theta_r\}$

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. で行った射影は同相写像であるため、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  $\mathbf{W}$  及び  $\mathbf{\Lambda}$  を用いて  $\mathbf{A}$  の固有値分解を計算する。具体的には、III. で述べた通り、 $\mathbf{A}$  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<sup>1</sup>。

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言う。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及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わかる。

$$\sigma_i = \frac{\text{Re}(\log(\lambda_i))}{\Delta t}$$

$$\omega_i = \frac{\text{Im}(\log(\lambda_i))}{\Delta t}$$

ここで、 $\text{Re}, \text{Im}$  はそれぞれ複素数の実部及び虚部である。

## 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。

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) と言う。

[定義 2] (予測ウィンドウ  $X^F$ ) 時刻  $t_c$  から  $l_s$  ステップ先の区間で与えられる  $X^F$  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  $l_p = t_e - t_f + 1$  の時、 $l_p$  ステップごとに  $X^F$  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  $X^C = \mathbf{X}[t_m : t_c]$  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  $X^F = \mathbf{X}[t_f : t_e]$  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。

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

但し、 $h$  は遅延時間を意味する。上記の行列の構造にある通り、 $\mathbf{X}_{\text{aug}}$  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。尚、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、第 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。

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

これらを考慮した線形動的システムを次のように定義する。

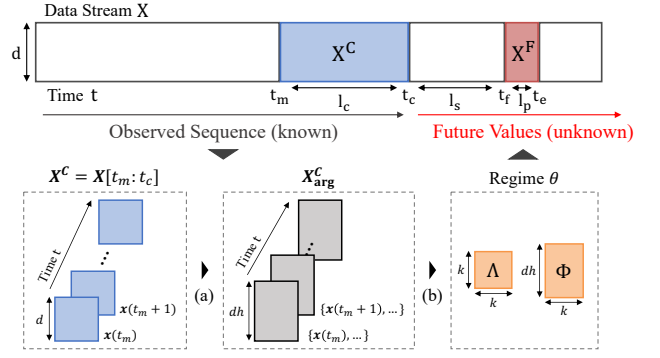


図 2 MODECAST のモデル概要図: (a) カレントウィンドウ  $X^C$  を遅延座標系へ射影し  $X^C_{\text{arg}}$  を得る。その後、(b)  $X^C_{\text{arg}}$  からレジーム  $\theta = \{\Lambda, \Phi\}$  を計算し、 $l_s$  ステップ先を予測する。

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

但し、初期条件は  $s(0) = \Phi^\dagger v(0)$  で与えられる。また、 $\Lambda, \Phi$  はそれぞれ動的モード分解で得られる固有値及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  $\theta$  は右の通りとする： $\theta = \{\Lambda, \Phi\}$ 。

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捉えたい。そのため時系列データストリームを複数のレジームで表現をしたい。そのために複数のレジームで構成されたレジームセット  $\Theta = \{\theta_1, \dots, \theta_r\}$  も同時に定義しておく。

提案モデルの全体図を図 2 に示す。提案モデルはカレントウィンドウ  $X^C$  に最も適したレジーム  $\theta = \{\Lambda, \Phi\}$  を遅延座標系及び動的モードを活用することで推定し、それを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  $X^C_{\text{aug}}$  に対して、 $P = (U^\top X^C_{\text{aug}_0} (U^\top X^C_{\text{aug}_0})^\top)^{-1}$  とする時、 $\omega = \{P, U\}$  を更新用パラメータと呼ぶ。但し、 $U$  は  $X^C_{\text{aug}}$  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  $r$  とした時、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  $s^c_{\text{out}}$  とする。これらをまとめて候補パラメータ  $C = \{\theta^c, \omega^c, s^c_{\text{out}}\}$  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  $\mathbf{C} = \{\theta^c, \omega^c, s_{out}^c\}$   
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のとおり (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  $\mathbf{M}$  及び候補パラメータ  $\mathbf{C}$  を更新する (Algorithm 2).
- MODEGENERATOR : 候補パラメータ  $\mathbf{C}$  が与えられた時,  $l_s$  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  $\mathbf{C}$  が与えられた時を考える. ここで, 損失関数を  $f(s_0^c, \theta^c, \mathbf{X}_{aug}^C)$  と定義する. 但し,  $s_0^c$  はレジーム  $\theta^c$  の初期条件であり,  $f(\cdot)$  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.,  $f(s_0^c, \theta^c, \mathbf{X}_{aug}^C) = \sum_{t=t_m}^{t_c} \|\mathbf{x}(t) - \mathbf{v}(t)\|$ ). 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時には, レジーム  $\theta^c$  に対して初期値  $s_0^c$  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  $s_0^c$  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  $s_0^c$  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.

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査し,  $\theta^c$  を更新する.

---

**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:  $\{s_0^c, \theta^c\} \leftarrow \arg \min_{\theta \in \Theta} \mathcal{F}(\theta, \mathbf{X}_{aug}^C)$   
6: **if**  $\mathcal{F}(\theta^c, \mathbf{X}_{aug}^C) > \rho$  **then**  
7: /\* Create new regime \*/  
8: Compute  $\Lambda, \Phi$  from  $\mathbf{X}_{aug}^C$  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:  $s_0^c \leftarrow g(\theta^c, \mathbf{X}_{aug}^C)$   
12: **else**  
13:  $\{\theta^c, \omega^c\} \leftarrow \text{REGIMEUPDATE}(\theta^c, \omega^c, \mathbf{X}_{aug}^C)$   
14: **end if**  
15: **else**  
16:  $s_0^c \leftarrow g(\theta^c, \mathbf{X}_{aug}^C)$   
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:  $\mathbf{C}' \leftarrow \{\theta^c, \omega^c, s_{out}^c\}$   
21: **return**  $\mathbf{M}', \mathbf{C}'$

---

(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  $\mathbf{V}_{aug}^C$  の終端の潜在値  $s_{out}^c$  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推定ウィンドウ  $\mathbf{X}_{aug}^C$  に未知のパターンが出現した時, 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第 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  $\mathbf{x}(t_c)$  及び 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'. 新たなデータ及びその一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.

$$\tilde{\mathbf{A}} \leftarrow (\mathbf{U}^\top \Phi) \Lambda (\mathbf{U}^\top \Phi)^\dagger$$

$$\tilde{\mathbf{A}} \leftarrow \tilde{\mathbf{A}} + \gamma (\tilde{\mathbf{x}}_1 - \tilde{\mathbf{A}} \tilde{\mathbf{x}}_0) \tilde{\mathbf{x}}_0^\top \hat{P}$$

但し,  $\gamma, \hat{P}$  とはそれぞれ以下の通り.

$$\hat{P} = \frac{P}{\delta}, \quad \gamma = \frac{1}{1 + \tilde{\mathbf{x}}_0^\top \hat{P} \tilde{\mathbf{x}}_0}$$

III'. 次の更新のために  $P$  を更新する.

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

IV'. 第3章の III. IV. の手順通りに更新後の  $\hat{A}$  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  $l_s$  ステップ先の予測に効果的な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. [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  $C$  が与えられた時を考える.

MODEGENERATOR は以下の手順で行われる.

(I) 式 (3) に従って,  $s_{out}^c$  を初期値として, 時刻  $t_e$  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カレントウィンドウ  $\mathbf{X}_{aug}^F$  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 512GB のメモリ, Xeon Gold 6258R 2.7GHz の 28 コア CPU および RTX A6000 48GB 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  $\epsilon = 0.5 \cdot \|\mathbf{X}^C\|$  とした.
- ARIMA: 古典的な時系列予測手法の一つ. 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

表 2 データセット詳細

データセット	属性・クエリ	次元
センサデータ: CMU Motion Capture Database [37] から取得		
(#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 [38] から取得		
(#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

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 [38] から毎週収集されたさまざまなクエリ (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第 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捉えることによって, 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捉え, 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 のリアルタイムに  $l_s = 100$  先の予測をしている様子及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及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 及び 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ヶ月先) と設定した. 全ての手法において, パラメータチューニングのためにシーク

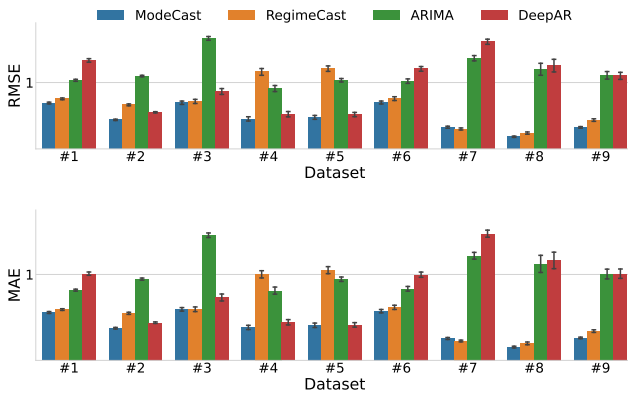


図3 MODECASTの平均予測精度の比較

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。図3は、多次元時系列データストリームにおけるMODECAST及び比較手法の予測誤差の平均値を示している。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。図に示す通り、提案手法は比較手法であるRegimeCast, ARIMA及び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捉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も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及び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が示す通り、MODECASTはデータストリームの長さに依存せず、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ためである。実際、図4が示す通り、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

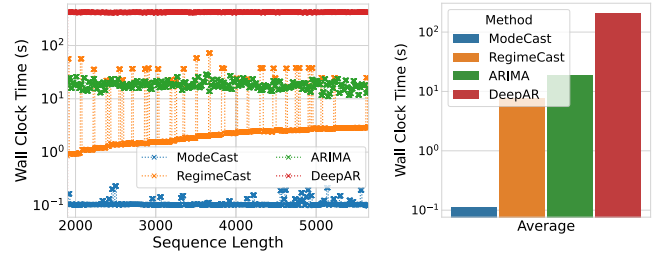


図4 各時刻  $t_c$  における計算コスト(左)と平均値(右): これらの図のy軸は対数スケールで表示されている。

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及び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言える。

## 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全て達成している。

- 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, JSTCRESTJPMJCR23M3, の助成を受けたものです。

## 文献

- [1] Gianmarco De Francisci Morales, Albert Bifet, Latifur Khan, Joao Gama, and Wei Fan. Iot big data stream mining. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2119–2120, 2016.
- [2] Mohammad Saeid Mahdavejad, Mohammadreza Rezvan, Mohammadamin Barekatin, Peyman Adibi, Payam Barnaghi, and Amit P Sheth. Machine learning for internet of things data analysis: A survey. *Digital Communications and Networks*, 4(3):161–175, 2018.
- [3] Koki Kawabata, Yasuko Matsubara, Takato Honda, and Yasushi Sakurai. Non-linear mining of social activities in tensor streams. In *Proceedings of the 26th ACM SIGKDD*

- International Conference on Knowledge Discovery & Data Mining*, pages 2093–2102, 2020.
- [4] Kota Nakamura, Yasuko Matsubara, Koki Kawabata, Yuhei Umeda, Yuichiro Wada, and Yasushi Sakurai. Fast and multi-aspect mining of complex time-stamped event streams. In *Proceedings of the ACM Web Conference 2023*, pages 1638–1649, 2023.
  - [5] Tasuku Kimura, Yasuko Matsubara, Koki Kawabata, and Yasushi Sakurai. Fast mining and forecasting of co-evolving epidemiological data streams. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3157–3167, 2022.
  - [6] Yasuko Matsubara, Yasushi Sakurai, Christos Faloutsos, Tomoharu Iwata, and Masatoshi Yoshikawa. Fast mining and forecasting of complex time-stamped events. In Qiang Yang, Deepak Agarwal, and Jian Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 271–279. ACM, 2012.
  - [7] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.
  - [8] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.
  - [9] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Number 9780199641178 in OUP Catalogue. Oxford University Press, 2012.
  - [10] Lei Li, B Aditya Prakash, and Christos Faloutsos. Parsimonious linear fingerprinting for time series. *Proceedings of the VLDB Endowment*, 3(1-2):385–396, 2010.
  - [11] Qiquan Shi, Jiaming Yin, Jiajun Cai, Andrzej Cichocki, Tatsuya Yokota, Lei Chen, Mingxuan Yuan, and Jia Zeng. Block hankel tensor arima for multiple short time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5758–5766, 2020.
  - [12] Spiros Papadimitriou, Anthony Brockwell, and Christos Faloutsos. Adaptive, hands-off stream mining. In *Proceedings 2003 VLDB Conference*, pages 560–571. Elsevier, 2003.
  - [13] Alysha M De Livera, Rob J Hyndman, and Ralph D Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American statistical association*, 106(496):1513–1527, 2011.
  - [14] Yasuko Matsubara and Yasushi Sakurai. Regime shifts in streams: Real-time forecasting of co-evolving time sequences. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1045–1054, 2016.
  - [15] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers.
  - [16] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.
  - [17] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.
  - [18] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.
  - [19] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.
  - [20] Yoshinobu Kawahara. Dynamic mode decomposition with reproducing kernels for koopman spectral analysis. *Advances in neural information processing systems*, 29, 2016.
  - [21] Naoya Takeishi, Yoshinobu Kawahara, Yasuo Tabei, and Takehisa Yairi. Bayesian dynamic mode decomposition. In *IJCAI*, pages 2814–2821, 2017.
  - [22] N Benjamin Erichson, Lionel Mathelin, J Nathan Kutz, and Steven L Brunton. Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(4):1867–1891, 2019.
  - [23] Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature communications*, 9(1):4950, 2018.
  - [24] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
  - [25] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
  - [26] Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005.
  - [27] Clarence W Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S Henningson. Spectral analysis of nonlinear flows. *Journal of fluid mechanics*, 641:115–127, 2009.
  - [28] N Benjamin Erichson and Carl Donovan. Randomized low-rank dynamic mode decomposition for motion detection. *Computer Vision and Image Understanding*, 146:40–50, 2016.
  - [29] Jacob Grosek and J Nathan Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.
  - [30] Jan Rohleff. 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] Joshua L Proctor and Philip A Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.
  - [32] Santosh Tirunagari, Samaneh Kouchaki, Norman Poh, Mirosław Bober, and David Windridge. Dynamic mode decomposition for univariate time series: analysing trends and forecasting. 2017.
  - [33] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pages 366–381. Springer, 2006.
  - [34] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis: proceedings of the biennial Conference held at Dundee, June 28–July 1, 1977*, pages 105–116. Springer, 2006.
  - [35] Hao Zhang, Clarence W Rowley, Eric A Deem, and Louis N Cattafesta. Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, 18(3):1586–1609, 2019.
  - [36] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
  - [37] Carnegie Mellon University Motion Capture Database. <http://mocap.cs.cmu.edu/>.
  - [38] Google Trends. <https://trends.google.com/trends/>.