情報処理学会論文誌　データベース　Vol.9　No.4　32–45 (Dec. 2016)

大規模データストリームの将来予測アルゴリズム

松原 靖子[1,†1,a)]　櫻井 保志[1,b)]

受付日 2016年6月9日, 採録日 2016年9月21日

概要：本論文では，大規模時系列データストリームのための高速予測手法である RegimeCast について述べる．RegimeCast は，センサデータや Web のアクセス履歴等，様々な時系列パターンから構成される大規模データストリームが与えられたとき，それらの中から重要な特徴や潜在的なトレンドを発見し，長期的かつ継続的に将来のイベント予測を行う．より具体的には，本研究では，自然界の生態系モデルにおけるレジームシフトの概念を拡張し，時系列イベントデータを適応型非線形動的システムとして表現することで，複雑な時系列パターンを柔軟に表現する．提案手法は (a) 大規模データストリームの中から，重要な特徴を発見し，(b) 刻々と変化していく潜在的な時系列パターン（すなわち，レジーム）を自動的かつ高速に認識することで，長期的なイベント予測を実現する．ここで，提案手法は (c) データストリームの長さに依存せず，(d) 各時刻において，最適なイベント予測値を推定する．実データを用いた実験では，RegimeCast が様々な時系列データストリームの中から特徴的なパターンを発見し長期的な予測を行うことを確認し，さらに，最新の既存手法と比較し大幅な精度，性能向上を達成していることを明らかにした．

キーワード：時系列データストリーム，非線形動的システム，将来予測

Real-time Forecasting of Co-evolving Time Sequences

Yasuko Matsubara[1,†1,a)]　Yasushi Sakurai[1,b)]

Received: June 9, 2016, Accepted: September 21, 2016

Abstract: Given a large, online stream of multiple co-evolving event sequences, such as sensor data and Web-click logs, that contains various types of non-linear dynamic evolving patterns of different durations, how can we efficiently and effectively capture important patterns? How do we go about forecasting long-term future events? In this paper, we present RegimeCast, an efficient and effective method for forecasting co-evolving data streams. RegimeCast is designed as an adaptive non-linear dynamical system, which is inspired by the concept of "regime shifts" in natural dynamical systems. Our method has the following properties: (a) *Effective*: it operates on large data streams, captures important patterns and performs long-term forecasting; (b) *Adaptive*: it automatically and incrementally recognizes the latent trends and dynamic evolution patterns (i.e., regimes) that are unknown in advance; (c) *Scalable*: it is fast and the computation cost does not depend on the length of data streams; (d) *Any-time*: it provides a response at any time and generates long-range future events. Extensive experiments on real datasets demonstrate that RegimeCast does indeed make long-range forecasts, and it outperforms state-of-the-art competitors as regards accuracy and speed.

Keywords: time series, non-linear dynamical systems, real-time forecasting

1. まえがき

時系列データストリームは，センサネットワーク監視 [16], [28], [34]，経済，産業分析 [47], [49]，ソーシャルネットワーク [20], [27]，オンラインテキスト [12], [13]，医療情報分析 [9], [15], [29] 等，多岐にわたる分野で大量に

[1] 熊本大学大学院先端科学研究部
Faculty of Advanced Science and Technology, Kumamoto University, Kumamoto 860–8555, Japan
[†1] 現在，国立研究開発法人科学技術振興機構，さきがけ
Presently with JST, PRESTO
[a)] yasuko@cs.kumamoto-u.ac.jp
[b)] yasushi@cs.kumamoto-u.ac.jp

生成される．これらの応用の中で，重要な要素技術としてあげられるのは，大量に生成され続ける時系列データのリアルタイム解析技術である．なかでも，時系列データストリームに基づくリアルタイム将来予測は，今後のビッグデータ時代における時系列解析技術として，最も重要かつ挑戦的な研究課題である．

本論文では，大規模時系列データストリームのための高速予測手法である RegimeCast [22] について述べる[*1]．RegimeCast は，自然界の生態系モデルにおけるレジームシフト [6], [43] の概念を拡張し，時系列イベントストリームを適応型非線形動的システムとして表現することで，複雑な時系列パターンを柔軟に表現する．

より具体的には，以下の問題を扱う．

d 次元のイベントエントリで構成される時系列データストリーム $X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c)\}$ が与えられ，t_c を現時刻とするとき，l_s ステップ先のイベント $\boldsymbol{x}(t_c + l_s)$ を予測する．

1.1　データストリームのリアルタイム予測

本研究では，d 次元のイベントエントリの集合で構成される半無限長のデータストリーム $X = \{\boldsymbol{x}(1), \boldsymbol{x}(2), \ldots, \boldsymbol{x}(t_c), \ldots\}$ を扱う．ここで，$\boldsymbol{x}(t_c)$ は最も新しいイベントエントリであり，時刻が進むごとに t_c が増加する．データストリーム X が与えられたとき，X の中から重要なパターンを発見し，未知の将来イベントをリアルタイムに推定したい．より具体的には，以下の要件を満たすような手法を提案する．

- **l_s ステップ先の予測**：複数のセンサから大量に生成される d 次元のイベントストリームとして，たとえば，自動車の走行センサや防犯用のモーションセンサ等を解析する場合を考える．これらの大規模イベントストリームが与えられたとき，効果的かつ効率的に未来のイベントを予測し，交通事故防止や防犯対策等を実現したい．ここで重要な点として，提案手法は，(a) **長期的な予測**の能力が必要である．たとえば，各時刻において，$l_s = 100$ ステップ先の推定をしたい．さらに，提案するアルゴリズムは (b) **連続性**を有し，データストリームの現在のトレンドや時系列パターンを動的に把握し，将来のイベントを継続的に予測し続けなくてはならない．

  より具体的に，まず，(a) **長期的な予測**については，たとえば，交通センサ解析に基づき「1 時刻先に事故が起こる可能性がある」といった警告を発した場合，その事故を防ぐには時間が短すぎる．つまり，リアルタイムの予測問題において，短期的な（たとえば，$1, 2, 3, \ldots$ ステップ先の）イベント予測は意味をなさない．次に，(b) **連続性**については，データストリーム処理において非常に重要な要素である．ARIMA モデル（autoregressive integrated moving average model）にあげられるような従来の静的な時系列予測手法と異なり，理想的な予測手法は，データストリームの最新時刻のトレンドやパターンに応じて柔軟かつ動的に推定値を最適化し続ける必要がある．

- **適応型非線形動的システム：図 1** (a) において示すとおり，現実世界における時系列イベントストリームは，種類の異なる様々な時系列パターンから構成される．たとえば，図 1 (a) における家事のパターンは walking（歩く），dustpan（塵取り），wipe a window（窓拭き）等の複数のモーションから構成される．同様に，ネットワーク監視システムで生成されるストリームは正常パターンに加え，異常値等の特徴的なパターンを含む．Web 上のユーザ活動（図 7）や商品購入履歴イベントストリームは，成長，成熟，衰退期等の複雑な非線形パターンから構成される．本論文では，このような特徴的な時系列パターンをレジーム（regime）と呼ぶ．図 1 (a) では，単一のイベントストリーム X の中に複数のレジームが含まれており，たとえば，$t = 1{,}000$ において，walking から dragging モーションへと移行している．ここで，このようなトレンドの急激な変化を**レジームシフト**と呼ぶ．

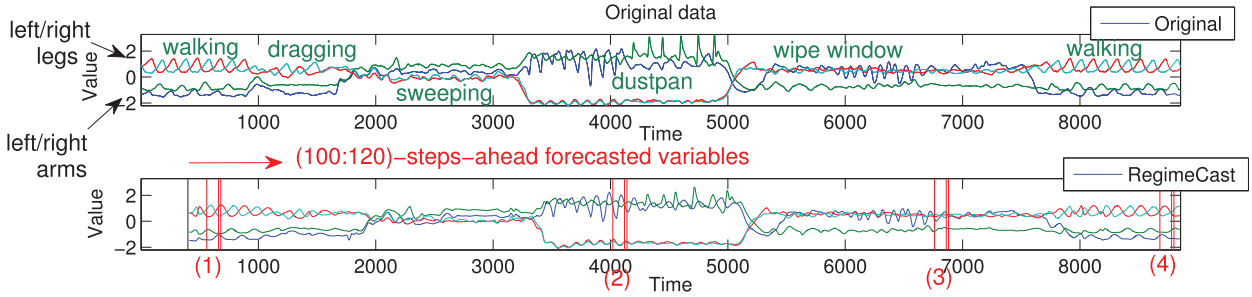
  まとめると，アルゴリズムはイベントストリームの中から突発的な変化点を検出し，現時刻のレジームを瞬時に認識することで，l_s ステップ先の未来のイベントを柔軟に予測しなくてはならない．ここでさらに重要な点として，これらのレジームは，事前知識なしに，データに応じて柔軟に学習する必要がある．そこで本研究では，適応型非線形動的システム（*adaptive* non-linear dynamical system）に基づく手法を提案する．提案モデルの詳細は 3.2 節において述べる．

1.2　具体例

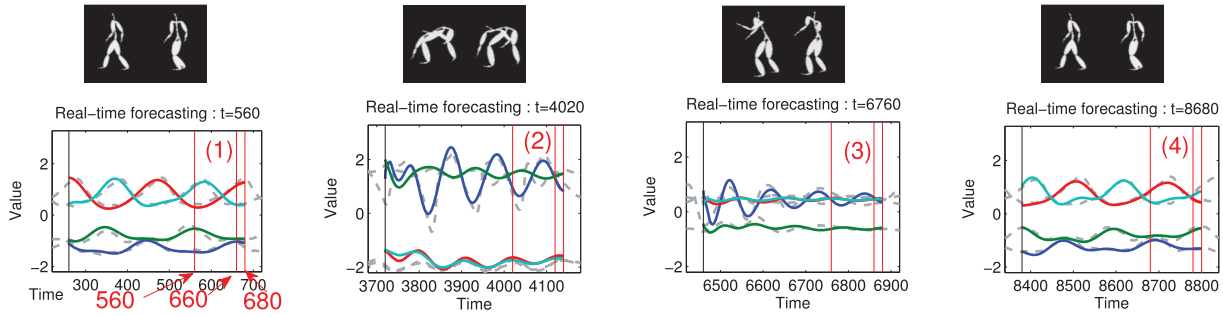
図 1 は，モーションイベントストリーム（house cleaning motion）における RegimeCast の出力の様子を示している．図 1 (a) 上段は，オリジナルのイベントストリームを示しており，4 つのモーションキャプチャセンサ（左右の腕および足）から生成された house cleaning に関連する複数のモーション（たとえば，walking，dragging a mop, using a dustpan, wiping a window）から構成される．図 1 (a) 下段は，RegimeCast を用いた際の，各時刻における (100:120) ステップ先の予測結果を示している．具体的には，RegimeCast は，出力単位時刻ごとに (100:120) ステップ先の将来イベントを予測し続けている[*2]．

[*1] http://www.cs.kumamoto-u.ac.jp/~yasuko/software.html

[*2] 出力単位時刻 l_p は，ユーザが設定するパラメータ（たとえば，$l_p = 20$）である．もし $l_p = 1$ の場合には，提案アルゴリズムは l_s ステップ先のイベントを 1 時刻ごとに出力する．



(a) オリジナルイベントストリーム（上段）と提案手法による (100:120)-ステップ先の予測結果（下段）



(b-1) Walking ($t_c = 560$)　(b-2) Dustpan ($t_c = 4020$)　(b-3) Wipe window ($t_c = 6760$)　(b-4) Walking ($t_c = 8680$)

図 1　モーションイベントストリームにおける RegimeCast の出力例：(a) オリジナルデータと提案手法による予測結果，(b) 各時刻における提案手法の出力結果の様子を示す

Fig. 1　Forecasting power of RegimeCast for a "house cleaning" motion stream: (a) The original data (top), our (100:120)-steps-ahead forecasted results (bottom), and (b) snapshots of video clips (top) and RegimeCast outputs (bottom) at four different time ticks.

図 1 (b-1)–(b-4) は，4 つの異なる時刻における RegimeCast の予測結果を示している．ここでは，それぞれ，図 1 (a) 下段に示した各時刻における予測結果のうち，時刻 $t_c = 560, 4020, 6760, 8680$ における出力の様子を示している．図 (b-1)–(b-4) において，上段は，各時刻における実際のモーションの様子，下段は RegimeCast の出力のスナップショットをそれぞれ示している．提案手法によって推定されたイベントは太線で表現され，オジリナルのイベントシーケンスは灰色の点線で表現されている．赤い縦線は時刻 $\{t_c, t_s, t_e\}$ を示しており，t_c は現時刻を示す．各時刻 t_c において，提案アルゴリズムは (100:120)-ステップ先の将来イベント，つまり，時刻 t_s から t_e を予測する．たとえば図 (b-1) では，現時刻 $t_c = 560$ において，提案手法は時刻 $t_s = 660$ から $t_e = 680$ を予測している．本論文では，図 (b-1)–(b-4) のようなスナップショット図を RegimeSnap と呼ぶ．より具体的な予測手法の内容については，4.1 節における図 3 で述べる．

まとめると，大規模イベントストリームが与えられたとき，提案手法は，高速かつ自動的に重要なモーションパターン（walking，wiping 等）を検出し，l_s ステップ先の将来イベントを予測する．最も重要な点として，提案アルゴリズムは，データに関する事前学習や，モーション（レジーム）の種類や出現数，変化点（レジームシフト）に関する事前情報を必要とせず，最適な予測値を高速に推定し続けることができる．

1.3　本論文の貢献

本研究では，大規模データストリームのための高速将来予測手法である RegimeCast を提案する．RegimeCast は次の特長を持つ．

(1)　大規模時系列ストリームの特徴的なパターンを発見し，長期的なイベントの予測を実現する．
(2)　データ内に含まれる時系列パターンの種類や変化点に関する事前情報を使用することなく，自動的かつ柔軟に潜在的なトレンドを認識する．
(3)　計算コストはデータストリームの長さに依存しない．
(4)　提案手法は，リアルタイムに最適な将来イベントを推定し続ける．

2.　関連研究

時系列データの解析に関する研究は多岐にわたる [2], [21], [30], [34], [39], [40], [41], [50]．大規模時系列シーケンスのための類似探索，パターン発見は重要な課題である [4], [31], [32], [34], [35], [38], [42], [44], [45]．時系列ビッグデータの研究としては，TriMine [26] は大規模複合時系列イベントデータのための高速な予測手法であり，FUNNEL [29] は大規模疫病テンソルデータのための非線形モデルである．文献 [23] では多次元時系列シーケンスの

ための特徴自動抽出手法を提案した．Rakthanmanon らは文献 [37] において，兆単位（“trillions”）の時系列シーケンスを対象とした DTW の類似探索問題を扱っている．

ソーシャルメディアとオンラインユーザ活動の非線形時系列解析に関する研究も活発化している [9], [24], [25], [27], [36]．文献 [27] では，ソーシャルネットワーク上での情報拡散過程を非線形動的システムとして表現し，Prakash ら [36] は，ネットワーク上において，2 つの異なる商品やアイディアがどのように競合するかを議論し，任意のグラフ構造上での理論的なモデル化を行った．Gruhl ら [10] はブログ等のオンライン活動と Amazon.com における売り上げの関係性に着目し，Ginsberg ら [9] は，オンライン検索数の推移からインフルエンザの流行をトラッキングし，実際のインフルエンザのウィルスとオンラインのユーザの活動に強い相関があることを示した．

自己回帰モデル（AR: autoregressive model），線形動的システム（LDS: linear dynamical systems），カルマンフィルタ（KF: Kalman filters）は代表的な技術であり，これらに基づく時系列の解析と予測手法が，AWSOM [33], TBATS [19]，PLiF [18]，TriMine [26] をはじめ，数多く提案されている．しかしながら，上記の予測手法はすべて線形方程式に基づくため，本研究で対象とする非線形性を有する時系列データの表現には適していない [39]．

既存の非線形の予測手法についても，主に最近傍探索に基づくものが主流であり [3], [48]，長期予測のための時系列のモデル化の能力を有さない．時系列データに含まれる複数のパターンを表現するモデルとして，階層的隠れマルコフモデル（hierarchical HMM）[5] や，swithcing LDS [7], BP-AR-HMM（beta process autoregressive HMM）[8] が提案されている．これらの手法は，時系列の複雑な動的パターンを表現する能力があるが，その一方で，高度なパラメータ学習やモデル構造の定義等が必要となり，さらに，大規模時系列データストリームのリアルタイム解析および将来イベント予測の能力を有していない．

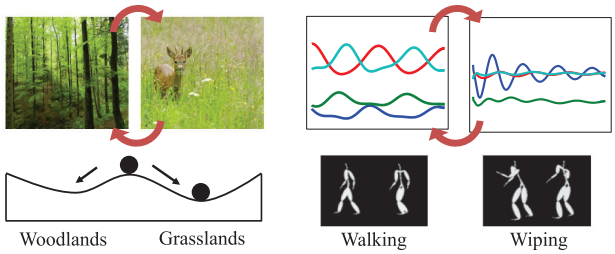
3. REGIMECAST

本章では，提案手法のための基本的な概念とモデルについて述べる．

3.1 自然界におけるレジームシフト

レジームシフトは，自然界における動的システムにおいて，構造や性質の急激な変化のことを指し，実世界の自然現象を理解するうえで重要な概念である．実社会における重要性により，レジームシフトは様々な分野において近年活発に研究されており，とりわけ，環境生態学分野において多く取り組まれる課題である [6], [11], [43], [46]．

環境生態学において，レジームとは，自然現象内の特徴的な時系列パターンのことを指し，レジームシフトとは，



(a) 生態系　(b) モーションセンサ

図 2　生態系とセンサストリームにおけるレジームシフトの様子
Fig. 2　Regimeshifts in an ecological system vs. sensor stream.

ある時系列パターン（レジーム）から別のレジームに変化する現象を示す [11]．レジームシフトは主に，内的要因（システム内の安定性の変化等）もしくは外部要因（システムへの外部ショック等）により引き起こされる．生態系において，たとえば，湖，珊瑚礁，森林地帯では，魚，珊瑚，水，土壌等の様々な生物的，非生物的な要素が，非線形動的システムとして相互作用し時間発展していく．自然生態系におけるレジームシフトの例として，人為的な富栄養化による湖の透明度の変化，海洋生態系におけるハリケーンや気候変化に基づく珊瑚と大型藻類の競合優位性の変化等があげられる．同様に，**図 2** (a) のように，草原地帯は森林地帯へシフトすることがある．たとえば草原地帯は，草食動物の存在や火事，森林伐採等の要素により木々の成長が抑えられ，安定したシステムとなるが，何らかの理由により木々が一定以上の大きさに成長すれば，草食動物や火事の影響を受ける可能性が減少し，森林地帯へ移行する [43] [*3]．

数理モデルに基づく分析． 生態系における各要素の時間発展は次の常微分方程式で表現することができる [43]：

$$\frac{ds(t)}{dt} = a_0 + a_1 s(t) + a_2 f(s(t)), \qquad (1)$$

ここで，$s(t)$ は時刻 t における生態系の特性（養分や土壌等）を表し，a_0 は栄養負荷等の $s(t)$ を変化させる環境要因を表現する．a_1 は，システム内の $s(t)$ の成長，減少率（たとえば $a_1 < 0$ における栄養除去率）を示す．a_2 は $s(t)$ の関数 $f(s(t))$ による回復率（栄養循環等）を示し，関数 f によりレジームの推移が発生する．

その簡易性と一般性により，上記の動的システムとレジームシフトの概念は，生態学をはじめ，社会学，経済学や政治学等において広く適応される．次節では，自然界の動的システムにおけるレジームシフトの概念を拡張し，提案手法である RegimeCast について述べる．

3.2 提案モデル

本節では提案モデルの詳細について述べる．本研究の目的は，様々な時系列パターン（つまりレジーム）を含む大規模イベントストリーム X が与えられたとき，その中か

[*3] Image courtesy of dan at FreeDigitalPhotos.net.

ら重要なトレンドを発見，モデル化し，長期的な予測を実現することである．

提案手法は次の 3 つの重要な特徴を表現する．

- (**P1**) 潜在的な非線形動的パターン
- (**P2**) イベントストリーム上でのレジームシフト
- (**P3**) 階層的な構造

3.1 節で示した自然界の動的システムと同様に，実世界におけるイベントストリームは，様々な潜在的要素に影響されながら時間発展していく．たとえば，走行センサストリームは，交通状況，天候，運転者等の要素により推移し，Web のアクセス履歴イベントは，ユーザの嗜好や興味に基づき時間発展する．そこで本研究では，(**P1**) 時系列イベントストリームの潜在的なパターンを非線形の動的システムとして表現する．より具体的には，時系列イベントシーケンスを潜在的非線形微分方程式として表現する．本研究ではさらに，重要な時系列パターンの変化点，つまり，(**P2**) イベントストリーム上でのレジームシフトを自動発見したい．図 2 は，生態系とイベントストリーム上（ここでは，モーションストリーム）でのレジームシフトを比較している．本研究では，生態系におけるレジーム間の変化，つまり，レジームシフトの概念に基づき，時系列イベントストリームを非線形動的システムとして表現することにより，複雑な時系列パターンを表現するモデルを提案する．ここで重要な点として，実際の時系列イベントストリームは，異なる時間発展に基づく多階層の動的システムから構成され，複雑な時系列パターンを有する．つまり，(**P3**) 階層的な構造をともなう．たとえば，図 7 で示すように，Web 上のユーザの活動は，10 年単位の成長，減衰のように長期的なパターンと，週，日，時間単位の短期的な時系列パターンから構成される．このような振舞いは自然界にも見られる．たとえば，珊瑚礁では，長期的な気候変化と突発的なハリケーン，双方からの影響を受ける場合がある [43]．そこで本研究では，より複雑で柔軟な時系列のモデル化と予測を実現するため，多階層構造に基づくモデル学習を提案する．

次に提案手法の詳細について述べる．

3.2.1　潜在的な非線形動的システム (P1)

まず最もシンプルな場合として，(**P1**) 単一の動的パターン（レジーム）の表現方法について述べる．つまり，ここでは，イベントシーケンスの中にレジームシフトは存在しないものとする．提案モデルは次の 2 種類の時系列活動パターンから構成される．

- $\boldsymbol{s}(t)$：潜在値，時刻 t における k 次元の潜在的な活動値（$\boldsymbol{s}(t) = \{s_i(t)\}_{i=1}^{k}$）．
- $\boldsymbol{v}(t)$：推定イベント，時刻 t における d 次元の観測値（$\boldsymbol{v}(t) = \{v_i(t)\}_{i=1}^{d}$）．

ここで，$\boldsymbol{v}(t)$ は時刻 t における実際のイベント観測値を示し（たとえば，d 個のセンサから生成される実測値），$\boldsymbol{s}(t)$ は，潜在的な時系列パターンを表現する．これにより，単一のレジームは次の式で表現される．

モデル 1　$\boldsymbol{s}(t)$ を時刻 t における k 次元の潜在値，$\boldsymbol{v}(t)$ を時刻 t における d 次元の推定イベントとする．単一のレジームは次の式で表現される．

$$\frac{d\boldsymbol{s}(t)}{dt} = \boldsymbol{p} + \mathbf{Q}\boldsymbol{s}(t) + \mathcal{A}\mathbf{S}(t) \tag{2}$$

$$\boldsymbol{v}(t) = \boldsymbol{u} + \mathbf{V}\boldsymbol{s}(t) \tag{3}$$

ここで，初期条件を $\boldsymbol{s}(0) = \boldsymbol{s_0}$，$d\boldsymbol{s}(t)/dt$ を時刻 t の導関数とし，$\mathbf{S}(t)$ を $\boldsymbol{s}(t)$ の 2 次形式の行列とする：$\mathbf{S}(t) = \boldsymbol{s}(t)^T\boldsymbol{s}(t)$．また，$\boldsymbol{p}$，$\mathbf{Q}$，$\mathcal{A}$ は潜在値 $\boldsymbol{s}(t)$ を生成するパラメータ集合であり，各成分が線形，指数，非線形の動的パターンを表現する[*4]．$\boldsymbol{u}$，$\mathbf{V}$ は時刻 t における潜在値 $\boldsymbol{s}(t)$ から推定イベント $\boldsymbol{v}(t)$ への射影を示す．さらに，非線形テンソル $\mathcal{A}$ については，動的システムの複雑化を防ぐために，スパースであることが重要である．詳細については 4 章において述べる．まとめると，以下を得る：

定義 1（単一レジームのパラメータ集合 $\boldsymbol{\theta}$）　$\boldsymbol{\theta}$ を単一の潜在的非線形動的システムにおけるパラメータ集合とする：$\boldsymbol{\theta} = \{\boldsymbol{s_0}, \boldsymbol{p}, \mathbf{Q}, \mathcal{A}, \boldsymbol{u}, \mathbf{V}\}$．

3.2.2　イベントストリームにおけるレジームシフト (P2)

次に，(**P2**) イベントストリーム上でのレジームシフトについて述べる．具体例として，図 2 (b) における $c = 2$ 種類のレジーム（walking，wiping）から構成されるモーションイベントストリームを考える．ここでは，異なる 2 種のモーション（walking，wiping）が任意のタイミングで交互に繰り返されるような，複雑なイベントを表現するため，順応性の高い時系列モデルが必要となる．

そこで本研究では，より複雑な時系列パターンを表現するために次の要素を導入する．

- $\boldsymbol{w}(t)$：レジーム活動値，時刻 t における c 個のレジームにおけるレジームシフトの推移値．

ここで，$\boldsymbol{w}(t)$ は時刻 t における i 番目のレジーム $(1 \leq i \leq c)$ の推移の強さを示す．これにより，モデル 1 を拡張し，次式を提案する．

モデル 2　$\boldsymbol{s}_i(t)$ を時刻 t における i 番目のレジームの潜在値，($\boldsymbol{s}_i(t) = \{s_{ij}(t)\}_{j=1}^{k}$)，$\boldsymbol{w}(t)$ を時刻 t における i 番目のレジームの強さ ($\boldsymbol{w}(t) = \{w_i(t)\}_{i=1}^{c}$)，$\boldsymbol{v}(t)$ を時刻 t における d 次元の推定イベントとする．提案モデルは次の式で表現される．

$$\frac{d\boldsymbol{s}_i(t)}{dt} = \boldsymbol{p}_i + \mathbf{Q}_i\boldsymbol{s}_i(t) + \mathcal{A}_i\mathbf{S}_i(t) \quad (i = 1, \ldots, c) \tag{4}$$

$$\frac{d\boldsymbol{w}(t)}{dt} = \boldsymbol{r}(t) \tag{5}$$

$$\boldsymbol{v}(t) = \sum_{i=1}^{c} w_i(t)\left[\boldsymbol{u}_i + \mathbf{V}_i\boldsymbol{s}_i(t)\right] \tag{6}$$

[*4] 本論文では，非線形動的パターンの要素 $\mathcal{A}$ を 2 次関数として扱う．

ここで，$d\boldsymbol{w}(t)/dt$ は時刻 t の導関数を示す．

モデル 2 において，新たなパラメータとして，$\boldsymbol{r}(t)$ を導入する．$\boldsymbol{r}(t)$ は，時刻 t における c 次元のベクトルとして表現される．$\mathbf{R}$ を，レジームシフトのダイナミクスを表現するパラメータ集合 $\mathbf{R} = \{\boldsymbol{r}(t)\}_{t=1}^{t_c}$ とする．ここで，t_c はイベントストリームの長さを表し，$\mathbf{R}$ をレジームシフト行列を呼ぶ．もし，イベントストリームが単一のレジーム（つまり $c = 1$）で構成されている場合には，提案モデルはモデル 1 と一致する．まとめると，以下を得る．

定義 2（レジームパラメータ集合 $\boldsymbol{\Theta}$） $\boldsymbol{\Theta}$ をレジームのパラメータ集合 $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c, \mathbf{R}\}$，とする．ここで，$c$ はイベントストリームに含まれるレジームの個数を示す．

3.2.3 階層構造 (P3)

これまでは，単一階層における動的システムについて述べた．しかしながら，先述のとおり，実世界における時系列イベントは，たとえば，Web 上のイベントにおける 10 年周期や 1 日周期のように，異なる時間発展に基づく時系列活動パターンを含んでいる．そこで，次の課題である (**P3**) 階層的な構造をともなう時系列パターンの表現のため，階層構造に基づくモデルを提案する．より具体的には，本研究では，複雑な時系列パターンを表現するために，多階層のレジーム集合 $\mathcal{M} = \{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots\}$ を用いる．階層 i においてローカルな推定イベント $\boldsymbol{v}^{(i)}(t)$ を生成し，重ね合わせることにより，実際の推定イベント $\boldsymbol{v}(t)$ を表現することができる．階層構造を用いたイベントの推定に関する具体的な方法については次章において述べる．

RegimeCast の全パラメータ集合． 提案モデルは次の要素で構成される．

定義 3（RegimeCast パラメータ集合：$\mathcal{M}$） $\mathcal{M}$ を階層構造をともなう時系列パターンを表現する提案モデルの全パラメータ集合とする：$\mathcal{M} = \{\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(h)}\}$．

4. アルゴリズム

本章では，大規模イベントストリームの予測手法である RegimeCast のアルゴリズムについて述べる．

4.1 問題定義

ここでは本手法で必要な概念について定義を行う．また**表 1** に主な記号と定義を示す．

定義 4（イベントストリーム：X） X を，d 次元のイベントエントリから構成されるデータストリーム $X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c)\}$ とし，t_c を現在の時刻とする．X をイベントストリームと呼ぶ．

ここで，毎時刻において新たなイベントエントリ $\boldsymbol{x}(t_c)$ が発生し，時刻が進むごとに t_c が増加するものとする．そこで，最新の時刻において発生したイベント集合をカレントウィンドウと呼び，次のように定義する．

定義 5（カレントウィンドウ：X_C） $X_C = X[t_m : t_c]$ を長さ l_c のカレントウィンドウとする．ここで，X_C はイベントストリーム X の時刻 t_m から時刻 t_c まで $(1 \le t_m \le t_c)$ の部分シーケンスを示す[*5]．

表 1　主な記号と定義
Table 1　Symbols and definitions.

記号	定義
d	時系列の次元数
t_c	現在の時刻
X	d 次元のイベントストリーム：$X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c)\}$
$\boldsymbol{x}(t)$	時刻 t における d 次元のイベント：$\boldsymbol{x}(t) = \{x_i(t)\}_{i=1}^{d}$
$\boldsymbol{s}(t)$	時刻 t における潜在値: $\boldsymbol{s}(t) = \{s_i(t)\}_{i=1}^{k}$
$\boldsymbol{w}(t)$	時刻 t におけるレジーム活動値: $\boldsymbol{w}(t) = \{w_i(t)\}_{i=1}^{c}$
$\boldsymbol{v}(t)$	時刻 t における推定イベント: $\boldsymbol{v}(t) = \{v_i(t)\}_{i=1}^{d}$
X_C	カレントウィンドウ: $X_C = X[t_m : t_c]$
V_F	予測ウィンドウ: $V_F = V[t_s : t_e]$
$c^{(i)}$	i 番目の階層におけるレジームの個数
$\boldsymbol{\theta}_j^{(i)}$	i 番目の階層におけるレジーム j のパラメータ集合
$\mathbf{R}^{(i)}$	i 番目の階層におけるレジームシフト行列
$\boldsymbol{\Theta}^{(i)}$	i 番目の階層の全パラメータ集合
$\mathcal{M}$	RegimeCast の全パラメータ集合：$\mathcal{M} = \{\boldsymbol{\Theta}^{(i)}\}_{i=1}^{h}$

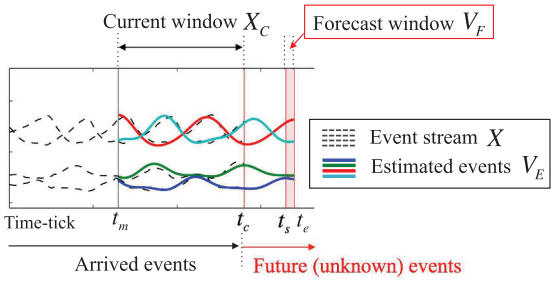


図 3　RegimeSnap の様子：提案手法は，オリジナルのストリーム X（黒点線）が与えられたとき，現時刻の時系列パターン V_E（色太線）を推定し，l_s ステップ先の将来イベント V_F（赤矩形内）を高速かつ継続的に出力する

Fig. 3　Illustration of RegimeSnap: Given a stream X, our algorithm estimates the current pattern V_E and reports future events V_F.

カレントウィンドウ X_C が与えられたとき，次の目標は，パラメータ集合 $\mathcal{M}$ の中から最適なレジームを発見し，モデル 2 に基づき l_s ステップ先の未来のイベント $V_F = \{\boldsymbol{v}(t_s), \ldots, \boldsymbol{v}(t_e)\}$ 推定することである．これを予測ウィンドウと呼ぶ．

定義 6（l_s ステップ先の予測ウィンドウ：V_F） $V_F = V[t_s : t_e]$ を l_s ステップ先の将来イベントシーケンス $(t_c \le t_s \le t_e)$ とし，$t_s = t_c + l_s$，$t_e = t_s + l_p$ とする．ここで，l_p は出力単位時刻の長さとする．

図 3 は，現時刻 t_c における RegimeCast のスナップショット（以下では RegimeSnap と呼ぶ）を示している．ここで，黒い点線はオリジナルのイベントストリーム X を示し，図の例では，$d = 4$ 次元のイベントエントリの集合

[*5] 本論文では $l_c = 3 \cdot l_s$ とする．

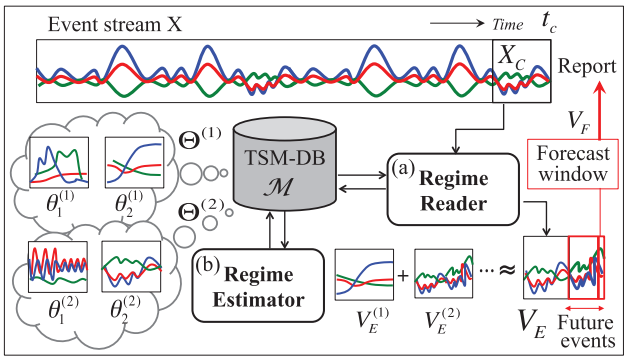


図 4　RegimeCast のアルゴリズム概要
Fig. 4　Overview of the RegimeCast algorithm.

で構成される．色のついた太線は，時刻 t_m から時刻 t_e における RegimeCast によるイベントの推定値 V_E を示す．ここで，時刻 t_c から時刻 t_e までの部分シーケンスは未来（つまり未知の）イベント集合であり，提案手法はこれらの時系列パターンを高速かつ継続的に推定し続けなくてはならない．

まとめると，イベントストリーム X が与えられたとき，本論文の目的は，カレントウィンドウ X_C に含まれる最新の時系列パターンを発見し，適応型非線形動的システムとして表現することで，l_s ステップ先の予測ウィンドウ V_F を高速かつ継続的に推定し続けることである．

問題 1　イベントストリーム $X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c), \ldots\}$ が与えられたとき，l_s ステップ先の将来イベント V_F を出力し続ける．より具体的には，各時刻 t_c において，

- カレントウィンドウ X_C に含まれる最適なレジームのパターンを検出し，
- X_C のレジームパターンに基づきモデルのパラメータ集合 $\mathcal{M}$ を更新し，
- l_s ステップ先の将来イベント V_F を出力する．

4.2　概要

RegimeCast は次のアルゴリズムで構成される．

- RegimeReader：カレントウィンドウ X_C とモデルパラメータ集合 $\boldsymbol{\Theta}$ が与えられたとき，レジームのダイナミクスを推定し，イベント $V_E = V[t_m : t_e]$ を生成する（Algorithm 1）．
- RegimeEstimator：カレントウィンドウ X_C の中に新たなレジームパターンが含まれていた場合に，X_C を表現する新たなレジームモデルパラメータ $\boldsymbol{\theta}$ を推定する（Algorithm 2）．
- RegimeCast：各階層 i（$i = 1, \ldots, h$）における最適なイベント集合 $V_E{}^{(i)}$ を推定し，推定イベント $V_E = V_E{}^{(1)} + V_E{}^{(2)} + \ldots$ を計算する．その後，l_s ステップ先のイベント（つまり V_F）を報告する．さらに，モデルパラメータ集合 $\mathcal{M}$ を更新する（Algorithm 3）．

図 4 は，RegimeCast の処理の流れを示す．イベントストリーム $X = \{\boldsymbol{x}(1), \ldots, \boldsymbol{x}(t_c)\}$ が与えられたとき，提案手法は毎時刻 t_c において，カレントウィンドウ X_C を取り出し，現時刻のレジームパターン V_E を推定し（$V_E = V_E{}^{(1)} + V_E{}^{(2)} + \ldots$），$l_s$ ステップ先のイベント V_F を出力する．さらに，必要に応じて過去のパターンが格納されている時系列モデルデータベース（TSM-DB: time-series model database）を更新する．

4.3　提案アルゴリズム

ここでは，アルゴリズムの詳細を述べる．議論の単純化のため，まずは，単一の階層（つまり，$h = 1$）のみを考慮し，単一のカレントウィンドウ X_C とレジームパラメータ集合 $\boldsymbol{\Theta}$ が与えられた場合に焦点を当てて議論を進める．

4.3.1　RegimeReader

時刻 t_c におけるカレントウィンドウ X_C とレジームパラメータ集合 $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c, \mathbf{R}\}$ が与えられた場合を考える．RegimeReader の目的は，図 4 (a) に示すとおり，現在のレジームパラメータ集合 $\boldsymbol{\Theta}$ に基づき，イベントシーケンス $V_E = V[t_m : t_e]$ を推定することである．ここで，適切なイベントの推定値 V_E を得るにはどうしたらよいだろうか．最も単純な解決法は，$\boldsymbol{\Theta}$ 内のパラメータ集合を固定し，モデル 2 に基づき $\boldsymbol{v}(t_m), \boldsymbol{v}(t_m + 1), \ldots$ を計算することである．しかしながら，実際のイベントストリームでは，カレントウィンドウ X_C に含まれる潜在的なトレンドは時間の経過とともに動的かつ連続的に変化していく．そこで本研究では，$\boldsymbol{\Theta}$ 内に含まれるレジームのパラメータを最新のカレントウィンドウ X_C のパターンに基づき最適化することを提案する．具体的には，提案アルゴリズムは，$\boldsymbol{\Theta}$ を X_C 内に含まれる現時刻の活動パターンに応じて $\boldsymbol{\Theta}$ 内のパラメータ集合を柔軟に更新していく必要がある．

Algorithm 1 は，RegimeReader の処理の流れを示している．RegimeReader は (I) 個々のレジームの最適化，(II) レジームシフトの同定の 2 つのパートから構成される．

(I) 個々のレジームの最適化． 個々のレジームパラメータ $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$（$i = 1, \ldots, c$）について，潜在値の初期状態 $\boldsymbol{s_0} \in \boldsymbol{\theta}_i$ を最適化する．具体的には，オリジナルイベントと推定イベントの 2 乗誤差を最小化（つまり $\min ||X_C - V_{C_i}||$）するような $\boldsymbol{s_0}$ を求める．ここで，関数 $f_C(\boldsymbol{s_0}|\boldsymbol{\theta})$ は，モデル 2 における，レジームパラメータ $\boldsymbol{s_0}$，$\boldsymbol{\theta}$ が与えられたうえでの推定イベント $V_C = \{\boldsymbol{v}(t_m), \ldots, \boldsymbol{v}(t_c)\}$ を示す．

(II) レジームシフトの同定． (I) で得られた c 個の推定イベントの集合 $\{V_{C_i}\}_{i=1}^{c}$ に基づき，時刻 t_c におけるレジームシフトの潜在的な動的パターンを推定する．具体的には，$\boldsymbol{\Theta}$ に含まれるレジームの集合を最適化するためにレジーム活動値 $\boldsymbol{w}(t_c)$ を推定し，式 (6) に基づき $\boldsymbol{\Theta}$ 内のレジームシフト行列 $\mathbf{R}$ を更新する（つまり $\min ||X_C - f_C(\boldsymbol{\Theta})||$）．その後，推定イベント $V_E = f_E(\boldsymbol{\Theta})$ をカレントウィンドウ X_C に対する最適値として計算する．

Algorithm 1 RegimeReader $(X_C, \mathbf{\Theta})$

Input: Current window X_C and current regime parameters $\mathbf{\Theta}$
Output: Estimated events $V_E = V[t_m : t_e]$ and updated regimes $\mathbf{\Theta}$
/* (I) Individual regime estimation */
for $i = 1 : c$ **do**
  /* Estimate $\boldsymbol{s}'_{\mathbf{0}}$ and activity V_{C_i}' for i-th regime $\boldsymbol{\theta}_i$ */
  $\{\boldsymbol{\theta}_i[\boldsymbol{s_0}], V_{Ci}\} = \arg\min_{\boldsymbol{s}'_{\mathbf{0}}, V_{C_i}'} ||X_C - V_{C_i}'||$; // $V_{C_i}' = f_C(\boldsymbol{s}'_{\mathbf{0}}|\boldsymbol{\theta}_i)$;
end for
/* (II) Estimate regime activity at current time tick t_c */
$\boldsymbol{w}(t_c) = \arg\min_{w_1,\ldots,w_c} ||X_C - \sum_{i=1}^{c} w_i V_{Ci}||$;
$\boldsymbol{r}(t_c) = \boldsymbol{w}(t_c) - \boldsymbol{w}(t_c - 1)$; // Calculate regime shift variable
$\mathbf{R} = \mathbf{R} \cup \boldsymbol{r}(t_c)$; $\mathbf{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_c, \mathbf{R}\}$; // Update full parameters
$V_E = f_E(\mathbf{\Theta})$; // Calculate estimated event V_E
return $\{V_E, \mathbf{\Theta}\}$;

Algorithm 2 RegimeEstimator (X_C)

Input: Current window X_C
Output: Estimated model parameter set $\boldsymbol{\theta} = \{\boldsymbol{s_0}, \boldsymbol{p}, \mathbf{Q}, \mathcal{A}, \boldsymbol{u}, \mathbf{V}\}$
/* Estimate linear dynamical parameters $\boldsymbol{\theta}_L = \{\boldsymbol{p}, \mathbf{Q}, \boldsymbol{u}, \mathbf{V}\}$*/
$\mathcal{A} = 0$; // Initialize tensor $\mathcal{A}$
$\{\boldsymbol{s_0}, \boldsymbol{\theta}_L\} = \arg\min_{\boldsymbol{s}'_{\mathbf{0}}, \boldsymbol{\theta}'_L} ||X_C - V_C||$; //$V_C = f_C(\boldsymbol{s}'_{\mathbf{0}}, \boldsymbol{\theta}'_L, \boldsymbol{\theta}_N)$
/* Estimate non-linear dynamical parameters $\boldsymbol{\theta}_N = \{\mathcal{A}\}$ */
$\{\boldsymbol{s_0}, \boldsymbol{\theta}_N\} = \arg\min_{\boldsymbol{s}'_{\mathbf{0}}, \boldsymbol{\theta}'_N} ||X_C - V_C||$; //$V_C = f_C(\boldsymbol{s}'_{\mathbf{0}}, \boldsymbol{\theta}_L, \boldsymbol{\theta}'_N)$
$\boldsymbol{\theta} = \{\boldsymbol{s_0}, \boldsymbol{\theta}_L, \boldsymbol{\theta}_N\}$; // Full parameter set
return $\boldsymbol{\theta} = \{\boldsymbol{s_0}, \boldsymbol{p}, \mathbf{Q}, \mathcal{A}, \boldsymbol{u}, \mathbf{V}\}$;

ここで，平均 2 乗誤差 $||\cdot||$ を最小化する方法として，本研究では，非線形性を有する学習に適した LM（Levenberg-Marquardt）アルゴリズム [17] を用いた．

4.3.2 RegimeEstimator

次に，新たなレジームを推定するためのアルゴリズムである RegimeEstimator（図 4 (b)）について述べる．ここでの課題は，カレントウィンドウ X_C に未知のレジームが含まれていた場合の処理についてである．提案アルゴリズムは，X_C に含まれる未知の時系列パターンを表現するため，新たなレジーム $\boldsymbol{\theta}$ を推定し，パラメータ集合 $\mathbf{\Theta}$ に挿入する．

ここで，重要な問題として，レジームを表現する $\boldsymbol{\theta}$ は非常に多くのパラメータ数から構成されている．一般に，非線形モデルにおける多数のパラメータの同時推定は，最適解の学習が非常に難しく，計算コストも高い．さらに，3.2.1 項で示したとおり，非線形活動テンソル $\mathcal{A}$ は単一のレジーム内の時系列パターンの複雑性を抑えるために，スパースであることが重要である．

そこで本研究では，線形，非線形の双方のパラメータ集合を高速かつ効果的に推定するためのアルゴリズムとして RegimeEstimator を提案する．具体的には，パラメータ集合 $\boldsymbol{\theta}$ を線形，非線形の 2 種の部分集合：$\boldsymbol{\theta}_L = \{\boldsymbol{p}, \mathbf{Q}, \boldsymbol{u}, \mathbf{V}\}$，$\boldsymbol{\theta}_N = \{\mathcal{A}\}$，に分割し，それぞれのパラメータ集合を個別に推定する．Algorithm 2 は，RegimeEstimator の詳細な挙動を示す．カレントウィンドウ X_C が与えられたとき，提案アルゴリズムは，まず非線形活動テンソルを $\mathcal{A} = 0$ とし，X_C の線形的なパターンを表現するための初期状態 $\boldsymbol{s_0}$，および，線形パラメータ集合 $\boldsymbol{\theta}_L$ を推定する．パラメータの推定には EM（expectation-maximization）アルゴリズム [1] を用いた．続いて，非線形要素 $\mathcal{A}$ に関し，LM アルゴリズムを用いて X_C と潜在値 V_C のエラー値を最小化するように最適化する．本研究では，非線形テンソル $\mathcal{A}$ について，モデルの複雑性を抑えるため，対角成分 $a_{ijk} \in \mathcal{A}$（$i = j = k$）のみを推定した．

4.3.3 RegimeCast

これまでは，単一階層のレジーム集合 $\mathbf{\Theta}$ に対する推定イベント V_E の生成方法について述べた．本研究の最終目的は，図 4 に示すとおり，多階層における時系列パターン $\mathcal{M} = \{\mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(h)}\}$ を表現し，l_s ステップ先の予測ウィンドウ V_F を推定することである．3.2.3 項で示したとおり，イベントストリーム X が与えられたとき，年，週，日単位等の様々な階層の動的パターンを表現したい．そこで本研究では，階層モデルに基づく予測手法を提案する．より具体的には，カレントウィンドウ X_C を h 個の階層的なイベント集合 $X_C = X_C{}^{(1)} + \cdots + X_C{}^{(h)}$ に分解することで，より効果的な予測を実現する．ここで，$X_C{}^{(i)}$ は i 番目（$i = 1, \cdots, h$）の階層におけるイベントを示し，次式で計算される：$X_C{}^{(i)} = g(X_C - \sum_{j=1}^{i-1} X_C{}^{(j)} | H(i))$ [*6]．関数 $g(\cdot|H(i))$ は，i 番目の階層における長さ $H(i)$ の移動平均を示す．

RegimeCast の詳細を Algorithm 3 に示す．時刻 t_c における新たなイベント $\boldsymbol{x}(t_c)$ が与えられたとき，提案手法は各階層 i におけるカレントウィンドウ $X_C{}^{(i)}$ を計算し，(I) イベントシーケンス $V_E{}^{(i)}$ を推定する．もし，$\mathbf{\Theta}^{(i)}$ 内に適切なレジームが存在しない場合（つまり，カレントウィンドウと推定イベントの誤差が ϵ 以上の場合[*7]），(II) 新たなレジーム $\boldsymbol{\theta}$ を生成し，レジームパラメータ集合 $\mathbf{\Theta}^{(i)}$ を更新する．最後に，(III) l_s ステップ先の予測ウィンドウ V_F を出力する．

ダイナミックポイントセット（DPS）に基づく高速化． 本研究におけるモデル推定手法である RegimeReader は，モデル 2 に示したとおり，複雑な動的システムに基づくため，各時刻 t_c において，潜在値 $S_E = \{\boldsymbol{s}(t_m), \ldots, \boldsymbol{s}(t_e)\}$ の推定に $\mathrm{O}(l_e)$ の計算量を要する．ここで，l_e は S_E の長さを示す．しかし，この計算時間はリアルタイム性を要する処理にはボトルネックとなりうる．

[*6] 本論文では $h = 2$，$H = \{2 \cdot l_s, 1\}$ とする．
[*7] 本論文では，$\epsilon = 0.5||X_C{}^{(i)}||$ とする．

Algorithm 3 RegimeCast ($\boldsymbol{x}(t_c)$)

Input: a new event $\boldsymbol{x}(t_c)$ at time tick t_c
Output: l_s-steps-ahead future events V_F
/* Initialize future window $V_F = 0$ */
for $i = 1 : h$ **do**
  Compute $X_C{}^{(i)}$; // Current window at i-level
  /* (I) Parameter fitting for regime activities */
  $\{V_E{}^{(i)}, \boldsymbol{\Theta}^{(i)}\} = \text{RegimeReader}(X_C{}^{(i)}, \boldsymbol{\Theta}^{(i)})$;
  /* (II) Regime estimation (if required) */
  $V_C{}^{(i)} = V^{(i)}[t_m : t_c]$; // Estimated events from t_m to t_c
  if $||X_C{}^{(i)} - V_C{}^{(i)}|| > \epsilon$ **then**
    $\boldsymbol{\theta} = \text{RegimeEstimator}(X_C{}^{(i)})$; $\boldsymbol{\Theta}^{(i)} = \{\boldsymbol{\Theta}^{(i)} \cup \boldsymbol{\theta}\}$;
  end if
end for
/* (III) l_s-steps-ahead future event generation */
$V_E = V_E{}^{(1)} + \cdots + V_E{}^{(h)}; V_F = V[t_s : t_e]$;
return V_F;

そこで本研究では，動的なイベント生成を高速化するための手法を提案する．具体的には，すべてのイベント集合 $S_E = \{\boldsymbol{s}(t_m), \boldsymbol{s}(t_m+1), \boldsymbol{s}(t_m+2), \ldots, \boldsymbol{s}(t_e)\}$ を生成するかわりに，S_E の部分集合である $\hat{S}_E = \{\boldsymbol{s}(t_m), \boldsymbol{s}(t_m+\delta), \boldsymbol{s}(t_m+2\delta), \ldots, \boldsymbol{s}(t_e)\}$ のみを生成する．ここで，部分集合 $\hat{S}_E$ をダイナミックポイントセット（DPS: dynamic point set）と呼ぶ．δ は，潜在値の時間の生成間隔（たとえば，$\delta = 0.1 \cdot l_s$）を示す．ダイナミックポイントセット $\hat{S}_E$ は，次に示す 4 次のルンゲ・クッタ法 [14] に基づき生成を行う．

$$\boldsymbol{s}(t+\delta) = \boldsymbol{s}(t) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) + O(\delta^5) \quad (7)$$

ここで，次のように定める：$d\boldsymbol{s}(t)/dt = F(\boldsymbol{s}(t))$，$K_1 = \delta F(\boldsymbol{s}(t))$，$K_2 = \delta F(\boldsymbol{s}(t) + \frac{1}{2}K_1)$，$K_3 = \delta F(\boldsymbol{s}(t) + \frac{1}{2}K_2)$，$K_4 = \delta F(\boldsymbol{s}(t) + K_3)$．これにより，モデル推定の計算時間が $\hat{S}_E$ の長さである $O(l_e/\delta)$ となり，飛躍的な高速化を実現することができる．

理論的な分析． l_e と l_c をそれぞれ，推定イベント集合 V_E の長さ，カレントウィンドウ X_C の長さ，δ をダイナミックポイントセットの値とし，c を $\mathcal{M}$ に含まれるレジームの総数（$c = \sum_{i=1}^{h} c^{(i)}$）とする．

補助定理 1　各時刻における RegimeCast の計算時間は最小で $O(c \cdot l_e/\delta)$，最大で $O(c \cdot l_e/\delta + l_c)$ となる．

証明 1　各時刻 t_c において，RegimeReader は c 個の最適なレジーム V_E を推定するために，$O(c \cdot l_e/\delta)$ の計算時間を要する．もし，カレントウィンドウ X_C に新たなレジームが含まれていた場合，RegimeEstimator は，パラメータ集合 $\boldsymbol{\theta}$ の推定に $O(l_c)$ を要する．したがって，最小で $O(c \cdot l_e/\delta)$，最大で $O(c \cdot l_e/\delta + l_c)$ の計算時間を要する．

5. 評価実験

本論文では RegimeCast 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　イベント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　イベントストリームの予測に対する計算時間の検証

実験は 32 GB のメモリ，Intel Core i7-3770K 3.50 GHz の CPU を搭載した Linux のマシン上で実施した．各データセットは平均値と分散値で正規化（z-normalization）して使用し，$k = 4$ とした．

5.1 Q1：提案手法の有効性

本節では，大規模イベントストリームに対する RegimeCast の予測能力を検証する．

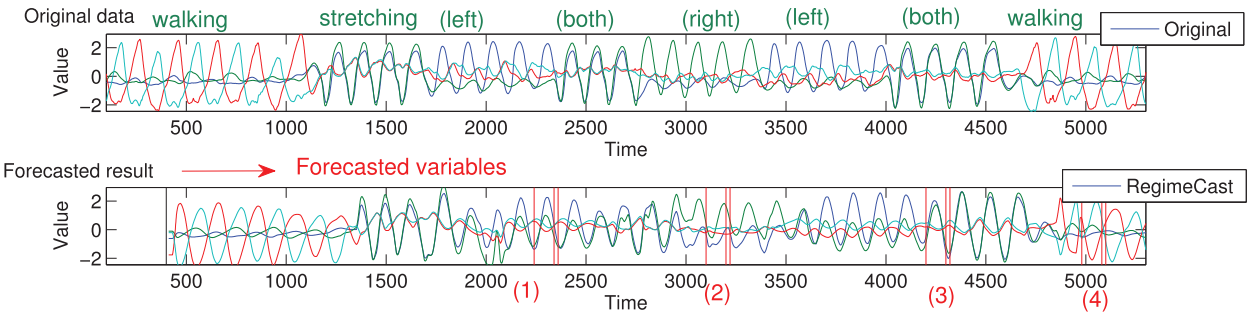
5.1.1 センサデータストリーム

図 1，**図 5**，**図 6** は，実際のモーションイベントストリームに対する提案手法の予測結果である．具体的には，それぞれ，*"house-cleaning"*，*"exercise"*，*"chicken-dance"* の 3 種類のイベントストリームに対する解析結果を示している．各データセットは左右の腕と足の動きから生成される $d = 4$ 次元のイベントシーケンスで構成され，それぞれのストリームは，walking や dancing 等の様々なモーションパターン（つまりレジーム）が含まれる[*8]．

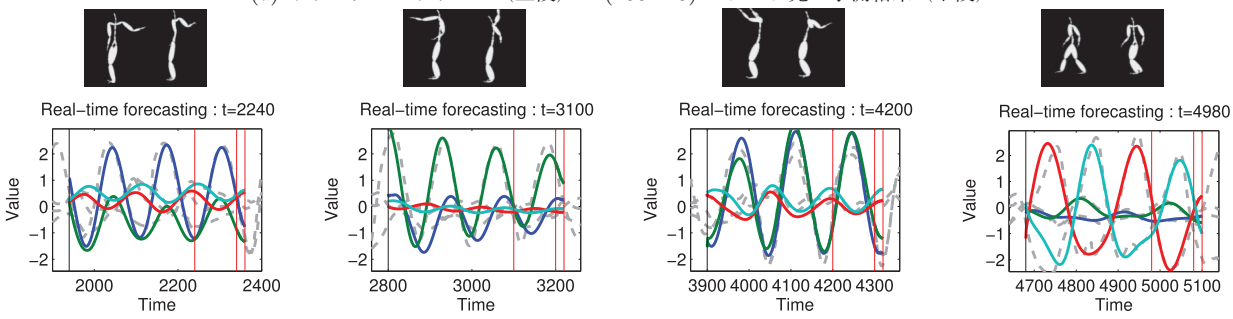
すでに 1 章の図 1 においても示したように，提案手法は自動的かつ効果的に，wiping モーションから walking モーションへの移り変わりをはじめとする複数のレジームシフトを検出し，長期的かつ継続的な将来イベントの予測に成功している．図 5 は，身体の運動に関するモーションストリームに対する RegimeCast の解析例を示している．図 5 (a) の上段はオリジナルデータを示し，下段は (100:120)-ステップ先の推定結果を示す．ここでは，出力単位時刻を $l_p = 20$ とした．より具体的には，RegimeCast は (100:120)-ステップ先の将来イベントを $l_p = 20$ 時刻ごとに出力する．たとえば，時刻 $t_m = 100$ に将来イベント $X[200 : 220]$ を，時刻 $t_m = 120$ に将来イベント $X[220 : 240]$ を，それぞれ出力している．図 5 (b) は，異なる 4 つの時刻における RegimeSnap の例を示す．図に示すとおり，RegimeCast は異なるレジームのパターンとその変化点，つまりレジームシフトを正しく検出していることが分かる．

図 6 はチキンダンス（chicken dance）に対する RegimeCast の予測結果を示している．具体的には，図 6 (a) の上段はオリジナルデータであり，beaks, wings, tail feathers, claps の 4 つの代表的なダンスステップから構成される．一般に，ダンスステップの動作は複雑な時系列パターンから構成されるため，予測が非常に難しい．具体的には，図 6 (b) に示しているように，チキンダンスにおける各ステップには，いくつかの基本的な動作が含まれている．たとえば，tail feathers というステップは，(a) 腕を素早く

[*8] http://mocap.cs.cmu.edu/



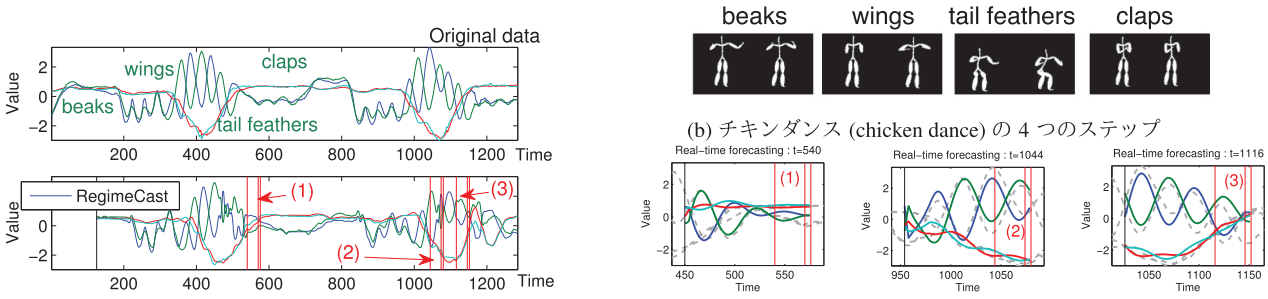
(a) オリジナルストリーム（上段）と (100:120)-ステップ先の予測結果（下段）



(b-1) Stretch/left ($t_c = 2240$)　(b-2) Stretch/right ($t_c = 3100$)　(b-3) Stretch/both ($t_c = 4200$)　(b-4) Walking ($t_c = 4980$)

図 5　モーションストリーム（xercise）対する RegimeCast の予測結果の様子 ($c^{(1,2)} = 27, 14$)

Fig. 5　Forecasting power of RegimeCast for the motion stream (*"exercise"*).



(a) オリジナルデータ (上段) と RegimeCast の予測結果 (下段)

(b) チキンダンス (chicken dance) の 4 つのステップ

(c-1) $t_c = 540$　(c-2) $t_c = 1044$　(c-3) $t_c = 1116$

図 6　モーションストリーム（chicken dance）に対する RegimeCast の予測結果の様子 ($c^{(1,2)} = 5, 9$)

Fig. 6　Real-time forecasting of RegimeCast for "chicken dance".

動かす，(b) 膝を 1 度曲げる，というテンポの異なる 2 つの基本動作から構成されており，多階層のレジームとして表現する必要がある．図 6 (a) の下段と (c-1)–(c-3) はそれぞれ，(30:35)-ステップ先の予測結果と，3 つの異なる時刻における RegimeSnap の様子を示している．図に示すように，提案手法は，複数の潜在的なレジームで構成される複雑な時系列パターンを表現し，長期的な動作の予測に成功している．ここで強調すべき点として，提案アルゴリズムは，事前知識やステップに関する情報を使用しない．RegimeCast は，重要な時系列パターン（レジーム）を高速に発見し，新たなレジームのパラメータを時系列モデルデータベースに格納することで，柔軟なイベント予測を継続的に行うことができる．

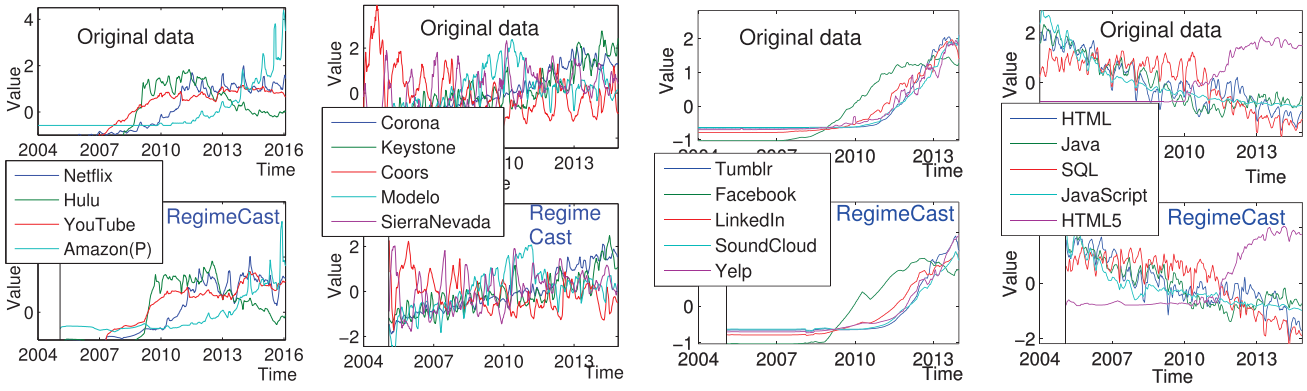
5.1.2　オンライン活動イベントストリーム

次に，Web 上のユーザ活動の予測について検証する．

図 7 は，*GoogleTrend* におけるイベントストリームの解析結果を示している．各データストリームは，Google[*9]における様々なクエリ（キーワード）の 2004 年から現在にかけての週単位の検索数で構成される．本論文では (a) オンライン *TV*，(b) ビール，(c) ソーシャルメディア，(d) ソフトウェアの 4 つのドメインに関するイベントストリームに対し，3 カ月先の予測を行った．

図 7 (a) は，オンライン *TV* に対する予測結果を示している．上段はオリジナルイベントを示し，Netfix ($\boldsymbol{x}_1$)，Hulu ($\boldsymbol{x}_2$)，YouTube ($\boldsymbol{x}_3$)，Amazon Prime ($\boldsymbol{x}_4$) の $d = 4$ 次元のイベントで構成される．近年，ビデオのストリーム配信サービスが急成長しており，RegimeCast はこれらのサービスの長期的な成長過程を柔軟に表現することができ

*9　*GoogleTrend*: http://www.google.com/trends/



(a) オンライン *TV*　(b) ビール　(c) ソーシャルメディア　(d) ソフトウェア

図 7　オンラインユーザ活動ストリームに対する 3 カ月先のイベント予測の様子：(a) オンライン *TV* ($c^{(1,2)} = 13, 5$)，(b) ビール ($c^{(1,2)} = 23, 1$)，(c) ソーシャルメディア ($c^{(1,2)} = 13, 1$)，(d) ソフトウェア ($c^{(1,2)} = 19, 3$)

Fig. 7　RegimeCast successfully forecasts 3-months-ahead future events of online user activities.

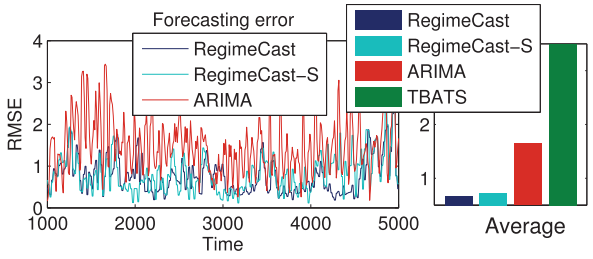
る．たとえば，2011 年から 2012 年にかけて，レジームシフトポイントが存在している．具体的には，Hulu（$\boldsymbol{x}_2$，緑線）は 2011 年から停滞，減少傾向にあるが，これはおそらく，Netflix（$\boldsymbol{x}_1$，青線）との間に潜在的な競合関係が存在しており，Hulu の顧客が Netflix へ興味を移したことに起因していると考えられる．提案手法である RegimeCast は，これらのレジームの変化点を自動的に検出し，柔軟かつ即座に未来のイベントを予測することができる．

図 7 (b) はビールに関するイベントストリームの予測結果を示しており，RegimeCast がビール産業に関する非線形の成長過程を柔軟に表現していることが分かる．各ビールの銘柄の検索数は，10 年間の間に著しく成長しているが，Coors（赤線）のみが例外となっている．Coors はアメリカ合衆国コロラド州に拠点を置くブランドである．同様にして，図 7 (c)，(d) に示すとおり，RegimeCast はソーシャルメディア，ソフトウェアにおける非線形パターンも正しく検出し，成長や競合，減少パターン等，様々なレジームを発見し，予測することに成功した．

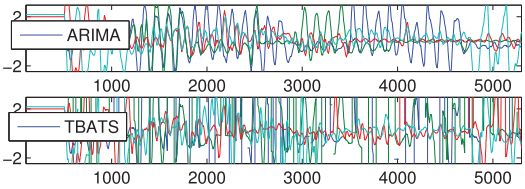
5.2　Q2：提案手法の精度

本論文では，RegimeCast の予測精度を検証するため，既存手法である (a) ARIMA，そして，最新の予測手法である (b) TBATS [19] との比較を行った．ここで，ARIMA のパラメータ数は AIC を用いて決定した．本実験ではさらに，提案手法における階層構造 (**P3**) の効果を検証するため，単一階層のみ（つまり $h = 1$，$H = \{1\}$）を用いて予測を行う場合の予測精度も検証した．これを RegimeCast-S と呼ぶ．

図 8 (a) は，モーションイベントストリーム（exercise，図 5 (a)）における RegimeCast の予測精度を示している．具体的には，図 8 (a) は，オリジナルデータと，(100:120)-ステップ先の予測イベントの推定値との二乗平均誤差 (RMSE: root mean square error) を示している．左図は各時刻における予測結果のエラー値（RMSE），右図は平均値を示す．ここで，左図における TBATS の結果は，エラー値がきわめて高いため省略した．図に示すとおり，RegimeCast は既存手法である ARIMA，TBATS および単一構造の RegimeCast-S と比較し，高い予測精度を持つ．図 8 (b) は，ARIMA と TBATS における実際の予測結果を示している．提案手法による予測結果（図 5 (a)）と比較し，既存の予測手法である ARIMA と TBATS は，非線形の時系列パターンとその変化点であるレジームシフトを表現できないため，適切に予測することができない．特に TBATS は大規模時系列データの中からのパターン発見に失敗し，



(a) 各時刻における予測値のエラー値（左）と平均値（右）



(b) ARIMA（上段）と TBATS（下段）による予測結果

図 8　モーションイベントストリーム（図 5 (a)）に対する RegimeCast の予測精度（RMSE）と既存手法との比較

Fig. 8　Forecasting error (RMSE) for the motion event stream.

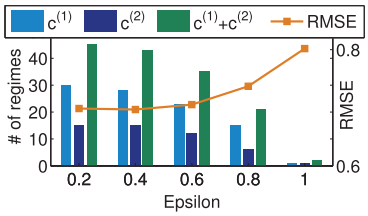
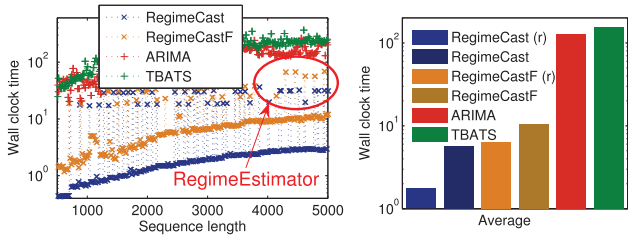
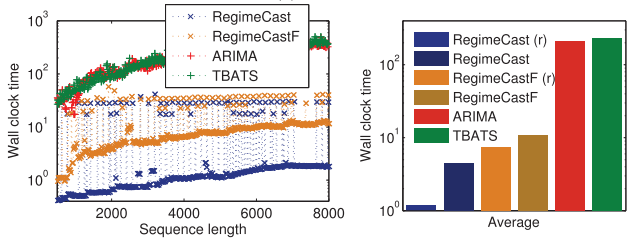


図 9　モーションイベントストリーム（exercise）に対するレジームの個数と予測精度の比較

Fig. 9　Forecasting accuracy vs. number of regimes.



(a) “Exercise”



(b) “House-cleaning”

図 10　各時刻 t_c に対する計算コスト（左）と平均値（右）

Fig. 10　Wall clock time vs. sequence length t_c (left) and average (right).

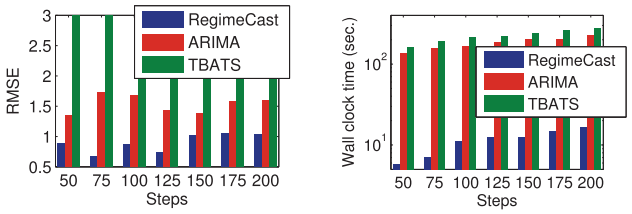
推定値が発散している．

図 9 は同データに対し，RegimeEstimator の閾値 ϵ を変化させた場合 $\epsilon = \alpha||X_C{}^{(i)}||$（$\alpha = 0.2, 0.4, \ldots, 1.0$）の予測精度（RMSE）とレジームの個数の関係を示している．図に示すとおり，より多くのレジームを学習すると，より高い精度で将来予測を行うことができる．

5.3　Q3：提案手法の計算時間

続いて，提案アルゴリズムの性能を検証する．**図 10** は，RegimeCast におけるイベントストリームの長さ t_c に対する計算コストを既存手法である ARIMA と TBATS と比較したものである．ここで，図中の y 軸は対数スケールで示している．4 章において提案したダイナミックポイントセット（DPS）の効果を検証するため，ここではさらに，提案手法の特別なものとして，時間間隔を $\delta = 1$ とした場合の手法である RegimeCast-F とも比較を行った．

図に示すとおり，RegimeCast は既存手法と比較し，長期的なイベント予測に対する大幅な性能向上を達成した．具体的には，TBATS と比較し最大 270 倍の高速化を実現している．図 10 (左) において，赤丸で囲まれた場所にいくつかのスパイクが見られるが，これは，イベントストリーム内に新たなレジームが出現したことによる RegimeEstimator の処理によるものである．図 10 (右) では，イベントストリーム全体の計算時間の平均値を示している．ここで，RegimeCast/RegimeCast-F (r) は，RegimeReader の平均計算時間，RegimeCast/RegimeCast-F は，RegimeCast の平均計算時間をそれぞれ表している．



(a) 予測精度 (RMSE)　　(b) 計算コスト

図 11　モーションイベントストリーム（exercise）に対する l_s ステップ先の (a) 予測精度と (b) 計算コストの比較

Fig. 11　l_s-steps-ahead forecasting over the motion stream (*“exercise”*).

l_s ステップ先のイベント予測． 1 章において述べたとおり，本論文の目標は，長期的なイベント予測である．そこで本節では最後に，予測するステップ数に応じてどのように結果が変化するかを検証する．**図 11** は，ステップ数を $l_s = 50, 75, \ldots, 200$ のように変化させた場合の予測精度と計算コストを示している．具体的には，予測イベントのエラー値（RMSE）と計算時間を既存手法と比較している．図に示すとおり，本研究の提案手法は，いずれのステップ数 l_s においても，精度，性能ともに向上していることが分かる．

6.　むすび

本論文では，大規模時系列イベントストリームのための高速予測手法である RegimeCast について述べた．RegimeCast は，自然界の生態系モデルにおけるレジームシフトの概念を拡張し，時系列イベントストリームを適応型非線形動的システムとして表現することで，複雑な時系列パターンを柔軟に表現し，長期的なイベント予測を実現する．

実データを用いた実験では，RegimeCast が様々な種類のイベントストリームに対し，複雑な非線形パターンや変化点を高速かつ継続的に発見し，長期的な将来予測を高精度に行うことを確認した．今後の課題として，様々な時系列イベントデータをより柔軟かつ長期的に表現するための高度なモデル学習や，最適な数のレジームを自動推定するための手法について検討していく予定である．

謝辞　本研究の一部は JSPS 科研費 JP15H02705, JP16K12430, JP26280112, JP26730060, JST さきがけおよび総務省 SCOPE（受付番号 162110003）の助成を受けたものです．

参考文献

[1] Bishop, C.M.: *Pattern Recognition and Machine Learning* (*Information Science and Statistics*), Springer (2006).

[2] Box, G.E., Jenkins, G.M. and Reinsel, G.C.: *Time Series Analysis: Forecasting and Control*, *3rd edition*, Prentice Hall, Englewood Cliffs, NJ (1994).

[3] Chakrabarti, D. and Faloutsos, C.: F4: Large-scale automated forecasting using fractals, *CIKM* (2002).

[4] Chandola, V., Banerjee, A. and Kumar, V.: Anomaly detection: A survey, *ACM Comput. Surv.*, Vol.41, No.3, pp.15:1–15:58 (2009).

[5] Fine, S., Singer, Y. and Tishby, N.: The hierarchical hidden markov model: Analysis and applications, *Machine Learning*, Vol.32, No.1, pp.41–62 (1998).

[6] Folke, C., Carpenter, S., Walker, B., Scheffer, M., Elmqvist, T., Gunderson, L. and Holling, C.S.: Regime shifts, resilience and biodiversity in ecosystem management, *Annual Review of Ecology, Evolution, and Systematics*, Vol.35, pp.557–581 (2004).

[7] Fox, E.B., Sudderth, E.B., Jordan, M.I. and Willsky, A.S.: Nonparametric bayesian learning of switching linear dynamical systems, *NIPS*, pp.457–464 (2008).

[8] Fox, E.B., Sudderth, E.B., Jordan, M.I. and Willsky, A.S.: Sharing features among dynamical systems with beta processes, *NIPS*, pp.549–557 (2009).

[9] Ginsberg, J., Mohebbi, M., Patel, R., Brammer, L., Smolinski, M. and Brilliant, L.: Detecting influenza epidemics using search engine query data, *Nature*, Vol.457, pp.1012–1014 (2009).

[10] Gruhl, D., Guha, R., Kumar, R., Novak, J. and Tomkins, A.: The predictive power of online chatter, *KDD*, pp.78–87 (2005).

[11] Hare, S. and Mantua, N.: Empirical evidence for North Pacific regime shifts in 1977 and 1989, *Progress in oceanography*, Vol.47, No.2000, pp.103–145 (2000).

[12] Hoffman, M.D., Blei, D.M. and Bach, F.R.: Online learning for latent dirichlet allocation, *NIPS*, pp.856–864 (2010).

[13] Iwata, T., Yamada, T., Sakurai, Y. and Ueda, N.: Online multiscale dynamic topic models, *KDD*, pp.663–672 (2010).

[14] Jackson, E.: *Perspectives of Nonlinear Dynamics*, Cambridge University Press (1992).

[15] Keogh, E.J., Chu, S., Hart, D. and Pazzani, M.J.: An online algorithm for segmenting time series, *ICDM*, pp.289–296 (2001).

[16] Letchner, J., Ré, C., Balazinska, M. and Philipose, M.: Access methods for markovian streams, *ICDE*, pp.246–257 (2009).

[17] Levenberg, K.: A method for the solution of certain nonlinear problems in least squares, *Quarterly Journal of Applied Mathmatics*, Vol.II, No.2, pp.164–168 (1944).

[18] Li, L., Prakash, B.A. and Faloutsos, C.: Parsimonious linear fingerprinting for time series, *PVLDB*, Vol.3, No.1, pp.385–396 (2010).

[19] Livera, A.M.D., 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).

[20] Mathioudakis, M., Koudas, N. and Marbach, P.: Early online identification of attention gathering items in social media, *WSDM*, pp.301–310 (2010).

[21] Matsubara, Y., Li, L., Papalexakis, E.E., Lo, D., Sakurai, Y. and Faloutsos, C.: F-trail: Finding patterns in taxi trajectories, *PAKDD*, pp.86–98 (2013).

[22] Matsubara, Y. and Sakurai, Y.: Regime shifts in streams: Real-time forecasting of co-evolving time sequences, *KDD*, pp.1045–1054 (2016).

[23] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: Autoplait: Automatic mining of co-evolving time sequences, *SIGMOD*, pp.193–204 (2014).

[24] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: The web as a jungle: Non-linear dynamical systems for co-evolving online activities, *WWW*, pp.721–731 (2015).

[25] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: Non-linear mining of competing local activities, *WWW* (2016).

[26] Matsubara, Y., Sakurai, Y., Faloutsos, C., Iwata, T. and Yoshikawa, M.: Fast mining and forecasting of complex time-stamped events, *KDD*, pp.271–279 (2012).

[27] Matsubara, Y., Sakurai, Y., Prakash, B.A., Li, L. and Faloutsos, C.: Rise and fall patterns of information diffusion: Model and implications, *KDD*, pp.6–14 (2012).

[28] Matsubara, Y., Sakurai, Y., Ueda, N. and Yoshikawa, M.: Fast and exact monitoring of co-evolving data streams, *ICDM*, pp.390–399 (2014).

[29] Matsubara, Y., Sakurai, Y., van Panhuis, W.G. and Faloutsos, C.: FUNNEL: Automatic mining of spatially coevolving epidemics, *KDD*, pp.105–114 (2014).

[30] Matsubara, Y., Sakurai, Y. and Yoshikawa, M.: Scalable algorithms for distribution search, *ICDM*, pp.347–356 (2009).

[31] Mueen, A. and Keogh, E.J.: Online discovery and maintenance of time series motifs, *KDD*, pp.1089–1098 (2010).

[32] Palpanas, T., Vlachos, M., Keogh, E. and Gunopulos, D.: Streaming time series summarization using user-defined amnesic functions, *IEEE Trans. Knowledge and Data Engineering*, Vol.20, No.7, pp.992–1006 (2008).

[33] Papadimitriou, S., Brockwell, A. and Faloutsos, C.: Adaptive, hands-off stream mining, *VLDB*, pp.560–571 (2003).

[34] Papadimitriou, S. and Yu, P.S.: Optimal multi-scale patterns in time series streams, *SIGMOD*, pp.647–658 (2006).

[35] Patel, P., Keogh, E.J., Lin, J. and Lonardi, S.: Mining motifs in massive time series databases, *Proc. ICDM*, pp.370–377 (2002).

[36] Prakash, B.A., Beutel, A., Rosenfeld, R. and Faloutsos, C.: Winner takes all: Competing viruses or ideas on fair-play networks, *WWW*, pp.1037–1046 (2012).

[37] Rakthanmanon, T., Campana, B.J.L., Mueen, A., Batista, G.E.A.P.A., Westover, M.B., Zhu, Q., Zakaria, J. and Keogh, E.J.: Searching and mining trillions of time series subsequences under dynamic time warping, *KDD*, pp.262–270 (2012).

[38] Sakurai, Y., Faloutsos, C. and Yamamuro, M.: Stream monitoring under the time warping distance, *ICDE*, pp.1046–1055 (2007).

[39] Sakurai, Y., Matsubara, Y. and Faloutsos, C.: Mining and forecasting of big time-series data, *SIGMOD, Tutorial*, pp.919–922 (2015).

[40] Sakurai, Y., Matsubara, Y. and Faloutsos, C.: Mining big time-series data on the web, *WWW, Tutorial* (2016).

[41] Sakurai, Y., Papadimitriou, S. and Faloutsos, C.: Braid: Stream mining through group lag correlations, *SIGMOD*, pp.599–610 (2005).

[42] Sakurai, Y., Yoshikawa, M. and Faloutsos, C.: Ftw:

Fast similarity search under the time warping distance, *PODS*, pp.326–337 (June 2005).
[43] Scheffer, M., Foley, J.A., Carpenter, S.R., Folke, C. and Walker, B.H.: Catastrophic shifts in ecosystems, *Nature*, Vol.413, No.6856, pp.591–6 (2001).
[44] Toyoda, M., Sakurai, Y. and Ishikawa, Y.: Pattern discovery in data streams under the time warping distance, *VLDB J.*, Vol.22, No.3, pp.295–318 (2013).
[45] Vlachos, M., Gunopulos, D. and Kollios, G.: Discovering similar multidimensional trajectories, *ICDE*, pp.673–684 (2002).
[46] Zelnika, Y.R., Meron, E. and Bel, G.: Gradual regime shifts in fairy circles, *PNAS* (2015).
[47] Zhao, Y., Sundaresan, N., Shen, Z. and Yu, P.S.: Anatomy of a web-scale resale market: A data mining approach, *WWW*, pp.1533–1544 (2013).
[48] Zhou, J. and Tung, A.K.H.: Smiler: A semi-lazy time series prediction system for sensors, *SIGMOD*, pp.1871–1886 (2015).
[49] Zhu, Y. and Shasha, D.: Statstream: Statistical monitoring of thousands of data streams in real time, *VLDB*, pp.358–369 (2002).
[50] Zoumpatianos, K., Idreos, S. and Palpanas, T.: Indexing for interactive exploration of big data series, *SIGMOD*, pp.1555–1566 (2014).

松原 靖子　（正会員）

2006 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科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 年熊本大学大学院自然科学研究科日本学術振興会特別研究員（PD）．2014 年より同大学院助教．この間，カーネギーメロン大学客員研究員．2016 年 12 月より国立研究開発法人科学技術振興機構さきがけ研究員．2016 年度日本データベース学会上林奨励賞，山下記念研究賞受賞．大規模時系列データマイニングに関する研究に従事．ACM，日本データベース学会各会員．

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 年熊本大学大学院自然科学研究科教授．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）等受賞．データマイニング，データストリーム処理，センサデータ処理，Web 情報解析技術の研究に従事．ACM，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　渡辺 陽介）