DEIM Forum 2023 4a-6-5

Webデータにおけるパターンの自動抽出

塚本　圭祐†　　村山　太一††　　天方　大地†††　　松原　靖子††　　櫻井　保志††
原　　隆浩†††

† 大阪大学工学部　〒 565–0871 大阪府吹田市山田丘 2-1
††† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1-5
†† 大阪大学産業科学研究所　〒 567–0047 大阪府茨木市美穂ヶ丘 8-1
E-mail: †{tsukamoto.keisuke,amagata.daichi,hara}@ist.osaka-u.ac.jp,
††{taichi,yasuko,yasushi}@sanken.osaka-u.ac.jp

あらまし　近年，インターネットの普及に伴い Web データが増加しており，多くの場面で活用されている．Web データは人々の動きや社会の動向を反映したソーシャルセンサの役割を持つことから，Web データの解析は人々の意思決定に対する重要な知見を提供することができる．Web データの解析においてパターン抽出は重要であるが，既存のパターン抽出手法の多くはセンサデータのような定間隔な時系列データを想定したものばかりである．つまり，Web データのような不定間隔にイベントが発生するデータには適していない．本研究では，Web 上における長期間のイベントデータから自動でパターンを抽出するアルゴリズムを提案する．提案アルゴリズムは，長期間のデータを意味のある部分系列に分解しパターン抽出を行うレジーム分類の手法と，イベントデータのモデリングに用いられる点過程の１つである Hawkes 過程の組み合わせで構成されている．また，レジームの数や変化点に関する事前知識を必要とせずにモデリングを行うことで，Web データを意味のあるパターンに分類することを実現する．人工データと Web データを用いて実験を行い，提案アルゴリズムの有効性を検証した．

キーワード　時系列解析，Hawkes 過程，Web データ，パターン抽出

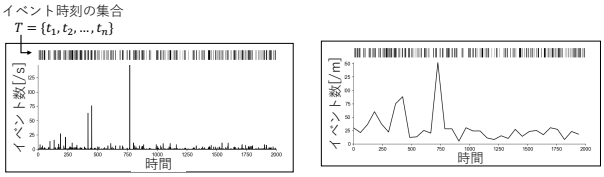
1　はじめに

近年，インターネットの普及に伴って Web データが増加しており，購買履歴を用いたユーザの購買行動予測 [12] やソーシャルメディアの投稿履歴を用いたインフルエンザの流行予測 [2] などの多くの場面で利用されている．Web データは人々の動きや社会の動向を反映したソーシャルセンサの役割を持つことから，Web データの解析は人々の意思決定に対する重要な知見を提供することができる．長期間の Web データは複数のパターンから構成される．例えば，ソーシャルメディアにおける，ある映画に関する投稿では，映画公開前の予告動画公開による投稿，映画公開後の映画の感想による投稿，および関連ニュースやグッズによる投稿などの様々なパターンが見られる．このようなパターンを抽出することで，Web における反応の変化を解析することができる．

Web データには，不定間隔にイベントが発生するデータ（イベントデータ）が多く存在する．例えば，Twitter や Reddit などのソーシャルメディアにおける投稿履歴や Web サイトにおけるアクセス履歴，EC サイトにおける購入履歴などがイベントデータとして蓄積されている．このようなデータをモデリングする手法として，点過程が頻繁に用いられており，特に Web データのような過去のイベントに影響してイベントが発生するデータには，点過程モデルの 1 つである Hawkes 過程を用いたモデリングが多く行われている [5, 19]．

時系列データのパターン抽出を行っている多くの既存研究では，センサデータのような定間隔な時間間隔で提供されるデータ（定間隔データ）を扱っている．Autoplait [15] では，隠れマルコフモデルの状態遷移をパターンごとにグループ化し，階層的な時系列レジームの遷移を表現している．TICC [8] では，各クラスタが多層マルコフ確率場を持つとして，時系列データのクラスタリングを行っている．このような手法はイベントデータをそのまま扱うことができないため，イベントデータに適用する際にはイベントデータを定間隔データに変換する必要がある．しかし，図 1(a) のように短い間隔で変換するとスパースなデータになり，図 1(b) のように長い間隔で変換するとイベント発生時刻および発生間隔の情報が欠落することから，適切なパターンを抽出することは難しい．

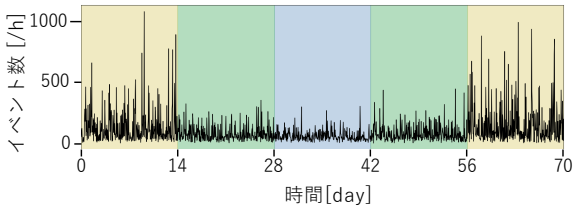
本研究では，長期間の Web データに対してパターンの変化点において分割を行い，各分割区間ごとにイベントデータをモデリングをすることで，自動でパターン抽出を行うアルゴリズムを提案する．図 2 はアルゴリズムの入力および出力の例である．提案アルゴリズムは，長期間のデータを意味のある部分系列に分解しパターン抽出を行うレジーム分類の手法と Hawkes 過程を組み合わせたモデルであり，分割区間ごとにモデリングを行うことで，Web データを意味のあるパターンに分類することを実現する．さらに，提案アルゴリズムはパターン（レジーム）の数や変化点に関する事前知識を必要とせずに，レジーム分類することが可能である．提案アルゴリズムでは，以下の 2 つのステップで構成されている．まず，Hawkes 過程の対数尤度



(a) 1秒あたりのイベント数　　(b) 1分あたりのイベント数

図 1: イベントデータから定間隔データへの変換

(a) 入力：イベント時刻の集合 $\mathcal{T} = \{t_1, ..., t_n\}$



(b) 出力：パターン抽出

図 2: 入力および出力の例

を用いて分割点を推定し，コスト関数で分割を行うか判定することによってイベントデータのセグメンテーションを行う．次に，分割されて新たにできた区間（セグメント候補）に対してレジームの推定を行う．具体的には，新しくできたセグメント候補に対して，既存レジームを当てはめることができるかを判定し，できない場合は新規レジームとして設定する．実験として，人工データを用いてレジーム分類の精度を評価し，提案アルゴリズムの有効性を検証した．また，Web データを用いて，提案アルゴリズムが適切なセグメンテーションおよびレジーム推定ができているかを確認した．

本論文の貢献は以下のとおりである．

- イベントデータに適応したモデルを提案
- 複数のレジームを自動で抽出
- Web データに対して実験を行い，モデル性能を検証

2 関 連 研 究

イベント発生時間のモデリング．本研究では，Web データにおける情報拡散のモデリングを行っている．このようなデータに関して，情報拡散を捉える点過程のモデルの 1 つである，自己励起性を考慮した Hawkes 過程 [10] が注目されている．Hawkes 過程は，地震 [17]，金融 [3,11]，購買 [21] およびソーシャルメディア [1,6] における拡散過程のモデル化に有効である．Hawkes 過程を拡張した研究は多く行われている．SEISMIC [22] および TiDeH [13] は，ソーシャルネットワークにおける再投稿の情報カスケードを Hawkes 過程を拡張してモデリングし，これまでの再投稿履歴から，最終的な再投稿数の予測を行う手法である．このモデルでは再投稿が行われる確率および投稿者のフォロワー数が考慮されている．最近では，ニューラルネットワークを用いて点過程のモデリングを行なった研究が多い．Recurrent Marked Temporal Point Process（RMTPP）[4] は，イベントの発生時刻にそのイベントに紐づいた情報を考慮するために深層学習を用いたモデルである．Dynamic Hawkes Process（DHP）[18] では，Hawkes 過程のカーネル関数の学習に現在のコミュニティの影響を考慮している．これらのモデルは 1 つのパターンが存在するデータのモデリングに適している．しかし，複数のパターンが存在するデータにおいては，複数のモデルを用いたモデリングが適している場合がある．例として，フェイクニュースの拡散を 2 種類の Hawkes 過程を用いてモデリングを行った研究 [16] が挙げられる．この手法は，フェイクニュースが普通のニュースとして拡散される段階とニュースをフェイクだと認識することで拡散される段階の 2 段階のプロセスで構成されることから，2 種類の Hawkes 過程を用いてカスケードのモデリングを行っている．しかし，この研究では 2 つのパターンが含まれたデータにのみ適用できるもので，さらに多くのパターンがあるデータには適用できない．そこで，本研究はパターンごとに複数の Hawkes 過程でモデリングを行なうことで，長期間のイベントデータに適用することを試みた．

時系列パターンの検出．センサなどから収集されるデータは，潜在的な特徴を持っている．その特徴の変化は，K-means [9]，線形力学系，および隠れマルコフモデルなどの手法で捉え，複数の時系列パターンに分類することができる．このような手法を拡張し，有用な潜在的な特徴を捉える研究が多く行われている．

- Parsimonious Linear Fingerprinting（PLiF）[14]：様々な時系列パターンを含む複雑なシーケンスに対して線形力学系を当てはめることで，意味のある特徴を抽出する手法．
- AutoPlait [15]：大規模時系列データのための時系列パターン自動抽出アルゴリズム．このアルゴリズムにおけるモデルは，各パターン間の遷移を表現するための多階層連鎖モデルおよび時系列パターンの発見のための最小記述長（MDL）原理 [7] に基づいた符号化スキームで構成されおり，パラメータの事前情報なしにパターン抽出を可能としている．
- Toeplitz Inverse Covariance-based Clustering（TICC）[8]: 各クラスタを時間的に変化しない部分列における異なるセンサ間の関係を示す依存関係ネットワークとし，時系列データのセグメンテーションとクラスタリングを同時に行う手法．ある時間における各特徴量をノード，特徴量間の関係をエッジとして捉えることで多層マルコフ確率場を形成し，これを各クラスタごとに持っていると仮定している．
- Time Adaptive Gaussian Model（TAGM）[20]：多次元時系列データに対して，隠れマルコフモデルとガウシアングラフィックモデルを組み合わせてデータマイニング，予測および因果関係パターンの検出を行う手法．多次元時系列の変数間の確率的関係を理解しながら，同時に時間方向にクラスタリングし，その基礎的な分布をグラフモデルにより推測することができる．

このように，時系列データからパターンごとにセグメンテーションを行い，潜在的な特徴を抽出する手法が多く研究されている．しかし，このようなモデルでは，定間隔な時間情報を持ったデータに適用される手法となっており，Web データのような不定間隔な時間情報を持ったイベントデータをそのまま適用することはできない．

3 事前準備

本章では，提案アルゴリズムの基礎となる手法について説明する．

3.1 点過程

点過程とは，空間上にランダムに分布する点の集合に関する確率過程である [23]．本研究では点をイベントとして捉え，Web データに適用する．観察期間 $[0, T]$ に n 個のイベントが発生したとき，イベント時刻の集合を $\mathcal{T} = \{t_1, t_2, ..., t_n\}$，$t_i \in [0, T]$ と表す．点過程では，強度 λ がイベントの発生しやすさを決めるパラメータである．強度が時間に応じて変化する場合は強度関数 $\lambda(t)$ として表される．微小な区間 $[t, t+\Delta t]$ にイベントが発生する確率を求めるために，強度関数が以下のように用いられる．

$$\begin{aligned} P[N(t, t+\Delta t) = 1] &= \lambda(t)\Delta t \\ P[N(t, t+\Delta t) = 0] &= 1 - \lambda(t)\Delta t \end{aligned} \tag{1}$$

また，強度関数は以下のように定義される．

$$\lambda(t) = \lim_{h \to 0} \frac{\mathbb{E}[N(t+h) - N(t)|H(t)]}{h} \tag{2}$$

ここで，$N(t)$ は時刻 t までのイベントの発生回数，$H(t)$ は時刻 t までのイベント履歴 $\{t_1, t_2, ...t_{N(t)}\}$ を表している．

3.2 Hawkes 過程

Web 上のイベントは，他のユーザが閲覧することで共有され，拡散される．このようなデータに対して，Hawkes 過程は過去のイベントに依存して次のイベントが発生するという特徴を持っており，現象を捉える上で最適なモデルとなっている．つまり，Hawkes 過程は一時的にイベントの発生確率が上昇する自己励起性を持ったデータを扱う点過程モデルである．Hawkes 過程の強度関数は以下のように定義される．

$$\lambda(t) = \lambda(t|H(t)) = \mu + \sum_{t_j < t} \phi(t - t_j) \tag{3}$$

ここで，μ は基底強度，$\phi(t)$ は過去のイベントからの影響を表すカーネル関数を表している．カーネル関数 $\phi(t)$ には，指数関数 $\alpha\beta e^{-\beta t}$ や冪関数 $\frac{K}{(t+c)^p}$ などが用いられ，Web データには指数関数が適していることから [1, 6]，本研究ではカーネル関数として指数関数を採用する．

$$\lambda(t) = \mu + \sum_{t_j < t} \alpha\beta e^{-\beta(t - t_j)} \tag{4}$$

また，このときの対数尤度は以下のようになる．

$$\begin{aligned} \mathcal{L} = \sum_{i=1}^{n} \log(\mu + \sum_{j<i} \alpha\beta e^{-\beta(t_i - t_j)}) \\ - (\mu T + \sum_{i=1}^{n} \alpha(1 - e^{-\beta(T - t_i)})) \end{aligned} \tag{5}$$

式 (5) が最大となるパラメータ集合 $\boldsymbol{\theta} = \{\mu, \alpha, \beta\}$ は，データを説明する最適な集合である．

3.3 レジーム分類

レジーム分類とは，入力時系列を m 個のセグメント集合 $\mathcal{S} = \{s_1, ...s_m\}$ に分割し，類似セグメントをグループごとに分類することである．例えば，図 2(b) では，複数のセグメントに分割され，各セグメントが { 青，緑，黄 } の 3 つのグループに分類されている．本研究ではこれらのグループをレジームと呼ぶ．s_i は i 番目のセグメントの開始点 t'_{is}，終了点 t'_{ie} で構成され（つまり，$s_i = \{t'_{is}, t'_{ie}\}$），各セグメントは重複がないものとする．本研究では，イベント時刻の集合 $\mathcal{T} = \{t_1, ..., t_n\}$ が与えられたとき，提案アルゴリズムは m 個のセグメント集合 $\mathcal{S} = \{s_1, ...s_m\}$ に分割してパターンの変動を捉え，レジーム分類をする．

定義 **1**（レジーム）それぞれのセグメント s はセグメントグループの 1 つに割り当てられる．これらのグループをレジームと呼び，レジームを表現するパラメータを $\boldsymbol{\theta}$ として表す．r を最適なセグメントグループの個数としたとき，r 個のレジームのモデルパラメータ集合を $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_r\}$ と表す．

さらに，各セグメントが所属するレジームを表現するため，新たにセグメントメンバシップを定義する．

定義 **2**（セグメントメンバシップ）i 番目のセグメントが所属するレジームの番号を f_i とし，m 個の整数列 $\mathcal{F} = \{f_1, ..., f_m\}$ として表す．

これにより，入力時系列を m 個のセグメントと r 個のレジームで $\{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ として表現することができる．

3.4 問題設定

本研究の目的は，イベント時刻の集合 $\mathcal{T} = \{t_1, t_2, ..., t_n\}$，$t_i \in [0, T]$ が与えられたときに，セグメンテーションを行い，各セグメントのレジーム分類を自動で行うこと，つまり，$\mathcal{T}$ を表現する最適なパラメータ集合 $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ を発見することである．

4 提案アルゴリズム

4.1 概　　要

提案アルゴリズムは，与えられたイベント時刻の集合 $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ に対してレジーム変化点の検出とレジームの推定を繰り返し，最終的に最適なセグメントとレジームの抽出を行う，

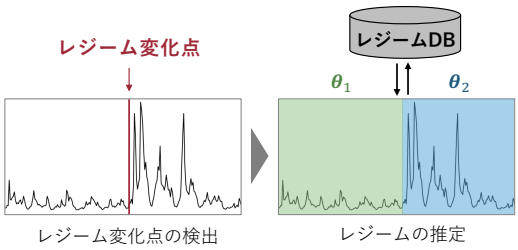


図 3: レジーム変化点の検出とレジームの推定の流れ

表 1: 主な記号と定義

記号	定義
n	イベント数
$\mathcal{T}$	イベント時刻の集合：$\mathcal{T} = \{t_1,..., t_n\}$
m	セグメントの数
s_i	セグメント:$s_i = \{t'_{is}, t'_{ie}\}$
$\mathcal{S}$	$\mathcal{T}$ に含まれるセグメント集合：$\mathcal{S} = \{s_1,...s_m\}$
$\mathcal{F}$	セグメントメンバシップ：$\mathcal{F} = \{f_1,..., f_m\}$
r	$\mathcal{T}$ に含まれるレジームの総数
$\boldsymbol{\Theta}$	r 個のレジームのモデルパラメータ集合：$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1,..., \boldsymbol{\theta}_r\}$
$\boldsymbol{\theta}_i$	i 番目のレジームのモデルパラメータ
$\mathcal{C}$	$\mathcal{T}$ を表現する最適なパラメータ集合:$\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$
$Cost_M(\boldsymbol{\Theta})$	$\boldsymbol{\Theta}$ のモデル表現コスト
$Cost_C(\mathcal{T}\|\boldsymbol{\Theta})$	$\boldsymbol{\Theta}$ による $\mathcal{T}$ のデータ符号化コスト
$Cost_T(\mathcal{T};\mathcal{C})$	判定コスト

つまり，$\mathcal{T}$ を表現する最適なパラメータ集合 $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ を発見する．各レジームを表現するモデルとして Hawkes 過程を採用し，Hawkes 過程のパラメータの集合 $\boldsymbol{\theta}_i = \{\mu_i,\alpha_i, \beta_i\}$ を保存する．各セグメント候補に対するレジーム変化点の検出とレジームの推定は以下のステップで行う．

- ステップ 1：レジーム変化点の検出（4.2 節）
- ステップ 2：レジームの推定（4.3 節）

上記アルゴリズムは，すべてのセグメント候補に適用していくもので，どのような順番で適用するかについては 4.4 節に記す．図 3 は 1 つのセグメント候補に対する，提案アルゴリズムの流れである．また，本研究における主な記号とその定義は表 1 に示す．

4.2 ステップ 1：レジーム変化点の検出

イベント時刻の集合 $\mathcal{T}$ のセグメント s_i にあたる部分集合 $\mathcal{T}_i = \{t \mid t'_{is} < t < t'_{ie}, t \in \mathcal{T}, s_i = \{t'_{is}, t'_{ie}\}\}$ を入力とし，レジーム変化点の候補を検出することで，分割するかどうかを判定する．このアルゴリズム（Algorithm 1）は分割候補点の検出と分割の判定の 2 つで構成されている．

4.2.1 分割候補点の検出

まず，$\mathcal{T}_i$ における最適な分割点を検出する．$\mathcal{T}_i$ を k 回分割し，各分割点において分割点前までの対数尤度 $\mathcal{L}_1$，分割点以降の対数尤度 $\mathcal{L}_2$ を計算する．$\mathcal{L}_1$ と $\mathcal{L}_2$ の和が最大となる点を分割候補点とする．分割候補点にて分割された前後のセグメントをそれぞれ s_f, s_b とする．

4.2.2 分割の判定

次に，分割候補点を採用するかを判定する．分割の判定基準として，最小記述長原理 [7] の概念に基づいた符号化コストを用いる．これはデータを圧縮すればするほど，その根底にあるパターンを知ることができるという仮定に基づいたものである．具体的には，分割によるイベント時刻の集合 $\mathcal{T}$ の符号化コストの減少量より，モデルを表現するために必要な符号化コストの増加量が上回る場合，データは圧縮されないため，分割を行わないこととする．

データが与えられたときのモデルのよさは次の式で表現できる．

$$Cost_T(\mathcal{T};\mathcal{C}) = Cost_M(\boldsymbol{\Theta}) + Cost_C(\mathcal{T}|\boldsymbol{\Theta}) \tag{6}$$

$Cost_T(\mathcal{T};\mathcal{C})$ を判定コストとする．ここで，$Cost_M(\boldsymbol{\Theta})$ は $\boldsymbol{\Theta}$ を表現するためのモデル表現コストであり，$Cost_C(\mathcal{T}|\boldsymbol{\Theta})$ は $\boldsymbol{\Theta}$ による $\mathcal{T}$ のデータ符号化コストである．$Cost_T(\mathcal{T};\mathcal{C})$ を最小化することで，最適なパラメータ集合 $\mathcal{C}$ を発見することができる．

モデル表現コストはモデル $\boldsymbol{\Theta}$ を記述するために必要なビット数であり，以下の要素で構成される．ここで，c_F は浮動小数点のコストを示す．

- セグメントの個数 m：$\log^*(m)$ ビット [1]
- レジームの個数 r：$\log^*(r)$ ビット
- セグメント集合 $\mathcal{S}$：$\sum_{i=1}^{m} \log^*(|s_i|)$ ビット
- レジームパラメータ集合 $\boldsymbol{\Theta}$：$3rc_F$ ビット
- レジーム割り当て $\mathcal{F}$：$m \log(r)$ ビット

データ符号化コストはデータを上手く表現できているかを表す指標である．レジーム $\boldsymbol{\theta}_j$ が与えられた際の，セグメント s_i のデータ符号化コストは，負の対数尤度（式 (5)) となる．つまり，モデル $\boldsymbol{\Theta}$ が与えられた際の，イベント時刻の集合 $\mathcal{T}$ のデータ符号化コストは，負の対数尤度の総和 $-\sum_{i=1}^{m} \mathcal{L}_i$ となる．

以上のことから，式 (6) は以下のようになる．

$$\begin{aligned} Cost_T(\mathcal{T};\mathcal{C}) = {} & \log^*(m) + \log^*(r) + m\log(r) + 3rc_F \\ & + \sum_{i=1}^{m} \log^*(|s_i|) - \sum_{i=1}^{m} \mathcal{L}_i \end{aligned} \tag{7}$$

分割候補点を採用するかを判定するために，2 つのセグメント s_f, s_b に分割した場合としていない場合の判定コストを比較する．分割した場合の判定コストが分割をしない場合の判定コストより小さいとき，分割を行う．

4.3 ステップ 2：レジームの推定

次に，分割点前後の 2 つのセグメント s_f, s_b それぞれのレジームを推定する．具体的には，各セグメントに対してレジームパラメータ集合 $\boldsymbol{\Theta}$ に保存されているレジームで表現可能であるか，もしくは新規レジーム $\boldsymbol{\theta}_{r+1}$ として保存するべきかを検証する．このアルゴリズム（Algorithm 2）は，セグメント

1：$\log^*$ はユニバーサル符号長を示す．

Algorithm 1 PatternChangeDetection

Input: Set of event times $\mathcal{T} = \{t_1, ..., t_n\}$,
Set of parameters $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$,
Segment candidate $s_i = \{t'_{is}, t'_{ie}\}$,
Candidate segment list $\mathcal{Q}_s$,
log-likelihood list $\mathcal{Q}_\mathcal{L}$
Output: Set of parameters $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$,
Candidate segment list $\mathcal{Q}_s$,
log-likelihood list $\mathcal{Q}_\mathcal{L}$

$\mathcal{T}_i \leftarrow \{t | t'_{is} < t < t'_{ie}, t \in \mathcal{T}\}$;
$\mathcal{P} \leftarrow$ *Cut point for dividing* t'_{is} *to* t'_{ie} *into* k *parts*;
for each $p_i \in \mathcal{P}$ **do**
  $\mathcal{T}_{i_1} \leftarrow \{t | t \leqq p_i, t \in \mathcal{T}_i\}$; $\mathcal{T}_{i_2} \leftarrow \{t | t > p_i, t \in \mathcal{T}_i\}$;
  $\mathcal{L}_{i_1} \leftarrow$ log-likelihood$(\mathcal{T}_{i_1})$; // Equation 5
  $\mathcal{L}_{i_2} \leftarrow$ log-likelihood$(\mathcal{T}_{i_2})$; // Equation 5
  if $\mathcal{L}_{i_1} + \mathcal{L}_{i_2} > \mathcal{L}_{max}$ **then**
    $\mathcal{L}_{max} \leftarrow \mathcal{L}_{i_1} + \mathcal{L}_{i_2}$; $p_{best} \leftarrow p_i$;
    $\mathcal{L}_f \leftarrow \mathcal{L}_{i_1}$; $\mathcal{L}_b \leftarrow \mathcal{L}_{i_2}$;
  end if
end for
$s_f \leftarrow \{t'_{is}, p_{best}\}$; $s_b \leftarrow \{p_{best}, t'_{ie}\}$;
Pop s_i from $\mathcal{S}$;
Push s_f, s_b into $\mathcal{S}$;
$\mathcal{C}' \leftarrow \{m+1, r+1, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$;
if $Cost_T(\mathcal{T}; \mathcal{C}) > Cost_T(\mathcal{T}; \mathcal{C}')$ **then** // Equation 7
  // Algorithm 2
  $\{r, \boldsymbol{\Theta}, \mathcal{F}\} \leftarrow$ RegimeInference$(\mathcal{T}, \{r, \boldsymbol{\Theta}, \mathcal{F}\}, s_f, s_b)$;
  $\mathcal{C} \leftarrow \{m+1, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$;
  Push s_f, s_b into $\mathcal{Q}_s$; Push $\mathcal{L}_f, \mathcal{L}_b$ into $\mathcal{Q}_\mathcal{L}$;
end if
return $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}, \mathcal{Q}_s, \mathcal{Q}_\mathcal{L}$

s_f, s_b で同一操作を行うため，分割点前のセグメント s_f を用いて説明する．

セグメント s_f に対して，新規レジーム $\boldsymbol{\theta}_{r+1}$ を用いた場合の対数尤度 (式 (5)) が，既存レジーム $\boldsymbol{\theta}_i \in \boldsymbol{\Theta}$ を用いた場合の対数尤度に対して閾値 γ 以上である場合，新規レジーム $\boldsymbol{\theta}_{r+1}$ を s_f にあたるレジームとし，レジームパラメータ集合 $\boldsymbol{\Theta}$ に加える．閾値 γ 未満である場合，対数尤度が最も大きくなる既存レジームをセグメント s_f にあたるレジームとする．

4.4 アルゴリズム

ここでは，イベント時刻の集合 $\mathcal{T}$ に対してステップ 1 およびステップ 2 を繰り返し適用するアルゴリズム（Algorithm 3）を示す．最初の入力としてイベント時刻の集合 $\mathcal{T}$ を 1 つのセグメント候補として扱い，各ステップを適用する．以後の入力は，最も対数尤度が小さいセグメント候補とし，各ステップを適用する．このとき，分割によって得られた 2 つのセグメント候補をセグメント候補リストに格納する．ただし，ステップ 1 において分割を行わないと判定したセグメント候補はセグメント候補リストから除き，最終的なセグメントとする．セグメント候補リストが空になったとき，アルゴリズムを終了する．

Algorithm 2 RegimeInference

Input: Set of event times $\mathcal{T} = \{t_1, ..., t_n\}$,
Part of parameters $\{r, \boldsymbol{\Theta}, \mathcal{F}\}$,
Candidate segments s_f, s_b
Output: Part of parameters $\{r, \boldsymbol{\Theta}, \mathcal{F}\}$

for each segment s_f, s_b **do**
  for $\boldsymbol{\theta}_j \in \boldsymbol{\Theta}$ **do**
    $\mathcal{L}_j \leftarrow$ log-likelihood$(\boldsymbol{\theta}_j, \mathcal{T}_i)$; // Equation 5
    if $\mathcal{L}_j > \mathcal{L}_{max}$ **then**
      $\mathcal{L}_{max} \leftarrow \mathcal{L}_j$; $\boldsymbol{\theta}_{best} \leftarrow \boldsymbol{\theta}_j$; $j_{best} \leftarrow j$;
    end if
  end for
  if $\mathcal{L}_{max} -$ log-likelihood$(\mathcal{T}_i) \geqq \gamma$ **then**
    $f_i \leftarrow j_{best}$;
  else
    Push $\boldsymbol{\theta}_{r+1}$ into $\boldsymbol{\Theta}$; $f_i \leftarrow r+1$;
    $r \leftarrow r+1$;
  end if
  Push f_i into $\mathcal{F}$;
end for
return $\{r, \boldsymbol{\Theta}, \mathcal{F}\}$

Algorithm 3 Proposed Algorithm

Input: Set of event time $\mathcal{T} = \{t_1, ..., t_n\}$
Output: Set of parameters $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$

$s_1 \leftarrow \{t_1, t_n\}$; $\mathcal{T}_1 \leftarrow \{t | t'_{1s} < t < t'_{1e}, t \in \mathcal{T}, s_1 = \{t'_{1s}, t'_{1e}\}\}$;
$\mathcal{L}_1 \leftarrow$ log-likelihood$(\mathcal{T}_1)$; // Equation 5
Push s_1 into $\mathcal{Q}_s$; // $\mathcal{Q}_s$: Candidate segment list
Push $\mathcal{L}_1$ into $\mathcal{Q}_\mathcal{L}$; // $\mathcal{Q}_\mathcal{L}$: log-likelihood list
while $\mathcal{Q}_s$ is not empty **do**
  $i \leftarrow$ index of minimum $\mathcal{L} \in \mathcal{Q}_\mathcal{L}$;
  // Algorithm1
  $\{\mathcal{C}, \mathcal{Q}_s, \mathcal{Q}_\mathcal{L}\} \leftarrow$ PatternChangeDetection$(\mathcal{T}, \mathcal{C}, s_i, \mathcal{Q}_s, \mathcal{Q}_\mathcal{L})$;
  Pop s_i from $\mathcal{Q}_s$; Pop $\mathcal{L}_i$ from $\mathcal{Q}_\mathcal{L}$;
end while
return $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$

5 実　　験

本章では，提案アルゴリズムの有効性を検証し，その結果を示す．ここでは，以下の 2 種類の実験を行った．

- 人工データを用いたレジーム分類精度の検証
- Web データを用いたケーススタディ

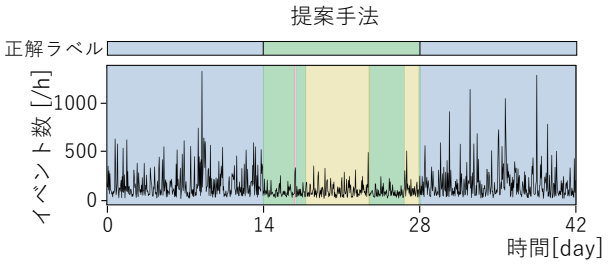
また，ステップ 1 における分割回数 k を 2000，ステップ 2 における閾値 γ をデータ期間の日数の 0.3 倍として実験を行った．

5.1 レジーム分類精度

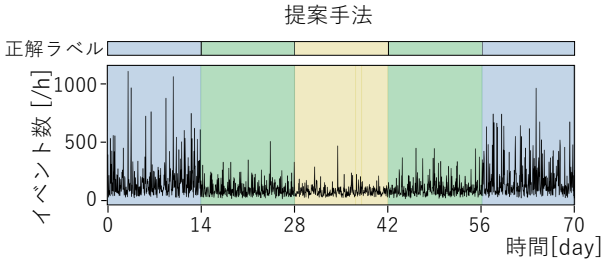
レジーム分類精度の検証を行うために，時系列パターンに関して正解ラベルを持つ人工のイベントデータを用いる．

5.1.1 評 価 指 標

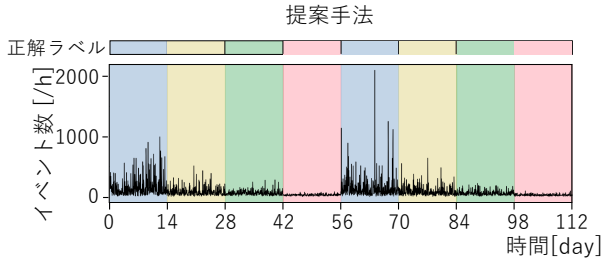
評価指標には，Macro F1-score を用いる．Macro F1-score は式（8）のように計算される．



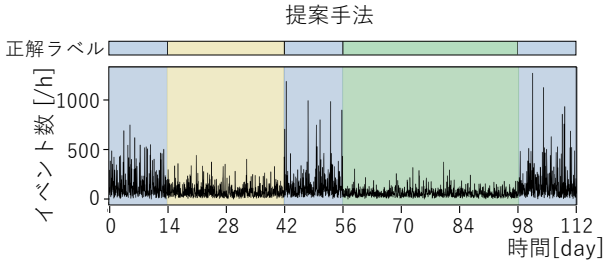
(a) 人工データ (i)



(b) 人工データ (ii)



(c) 人工データ (iii)



(d) 人工データ (iv)

図 4: (i), (ii), (iii), (iv) に提案手法および Autoplait を適用したレジーム分類結果

$$Macro\ F1\text{-}score = \frac{1}{N}\sum_{k}^{N}\frac{1}{\frac{1}{precision_k}+\frac{1}{recall_k}} \qquad (8)$$

ここで，N は正解ラベルのクラス数を示す．

5.1.2 比 較 手 法

比較手法では定間隔データを入力として扱うため，イベントデータを入力として扱えない．本実験では，イベントデータを定間隔データ（1 時間あたりのイベント数）に変換したデータを入力とした．提案手法の有効性を評価するために用いた比較手法は以下の通りである．

- K-means [9]：ユークリッド距離を用いた古典的なクラスタリング手法．クラスタ数は事前に指定されている．
- Autoplait [15]：レジーム間の遷移を表現するために隠れマルコフモデルを多階層連鎖的に拡張した手法．レジーム数やレジーム変化点などの事前情報なしにパターン抽出を行う．

5.1.3 データセット

提案アルゴリズムの性能を検証するために人工データを作成した．4 種類の異なる Hawkes 過程のパラメータ $\{\mu,\alpha,\beta\}$（表 2）から生成される複数のイベントデータを作成し，生成した複数のイベントデータを異なる組み合わせで結合することで，評価用の人工データを作成する．各イベントデータの日数は 14 日とする．イベントデータの組み合わせは [8] に従い，(i) “ 1,2,1 ”, (ii) “ 1,2,3,2,1 ”, (iii) “ 1,2,3,4,1,2,3,4 ”, (iv) “ 1,2,2,1,3,3,3,1 ” とした．さらに，ノイズとして全体のイベント数の 10 パーセントになるように全区間を対象にした一様分布から乱数を作成し，評価用の人工データに加えた．

表 2: 4 種類の Hawkes 過程のパラメータ

イベントデータ	μ	α	β
1	0.003	0.93	0.04
2	0.003	0.87	0.1
3	0.003	0.82	0.01
4	0.003	0.6	0.5

5.1.4 結　　果

各データセットでの Macro F1-score を表 3 に示す．また，図

4 は，(i)，(ii)，(iii)，(iv) に対して提案手法と Autoplait を適用したレジーム分類結果の例である．ここで，K-means を用いたレジーム分類ではレジームの切り替わりが激しく，多量のセグメントが生成されたため，図示しない．

提案手法は，レジーム分類結果の図から (ii)，(iii)，(iv) においてレジームの変化点を捉えて適切にセグメンテーションおよびレジーム推定ができたことがわかる．しかし，(i) においては両端のセグメンテーションは適切に行われているが，14 から 28 の区間では正解ラベルは 1 つのセグメントであるのにもかかわらず複数個のセグメントに分割された．これは，イベント数が少ない区間であることからノイズの影響を大きく受けたことや Hawkes 過程からの人工データの生成は確率的であることから区間内でばらつきができてしまったことが考えられる．提案手法は事前にレジーム数を指定せずに分割をしているため，このようなばらつきを新たなレジームに分類していた．比較手法では，イベントデータを定間隔データに変換することでイベント時刻やイベント発生間隔の詳細な情報が欠落しており，イベントの変化を適切に捉えることができず，評価指標の精度が低くなった．特に，自己励起性をもったイベントデータはイベント数が密な区間と疎な区間が短い間隔で繰り返されるデータであることから，イベント時刻やイベント発生間隔の情報の欠落が大きく影響していたと考えられる．

表 3: Macro F1-score によるレジーム分類精度

手法	(i)	(ii)	(iii)	(iv)
K-means	0.375	0.331	0.165	0.335
Autoplait	0.434	0.606	0.410	0.565
提案手法	0.750	0.997	0.999	0.999

5.2 ケーススタディ

提案アルゴリズムを Web データに適用した結果を示すことで，本手法の有効性を検証した．

5.2.1 データセット

数ヶ月にわたって収集されたハッシュタグが付与されたツイートの投稿履歴を用いる．データの詳細は表 4 に記す．

表 4: Web データセット

ハッシュタグ	期間	データ数
#ウルトラマン	2022/9/10–11/22	232,819
#麻生太郎	2022/5/8–9/12	20,233
#ショートケーキの日	2022/8/3–11/23	5,731

5.2.2 結　　果

#ウルトラマンのデータセット（図 5）では，赤色のレジームは，動画配信サイトにおいて新作映画（シン・ウルトラマン）の配信が決定した時間と配信が開始した時間であり，同じ影響による投稿を同じレジームとして捉えられている．配信開始後は，配信に関する感想が多くツイートされており，水色のレジームとして現れている．また，スマホゲームとのコラボやグッズの販売など，大きな影響が発生した際には黄色のレジーム，通常の時間帯は緑色のレジームとなっている．

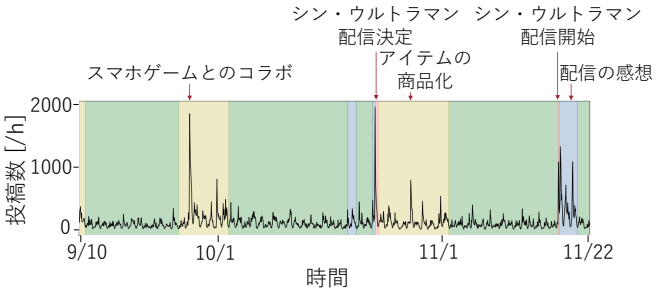


図 5: #ウルトラマン を含むツイートの投稿時間

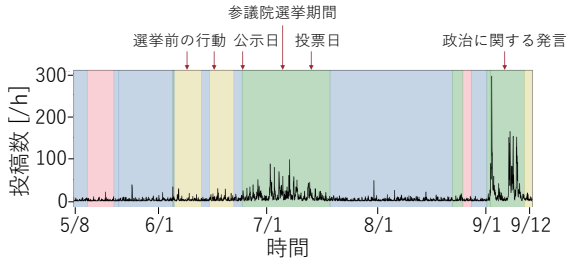


図 6: #麻生太郎 を含むツイートの投稿時間

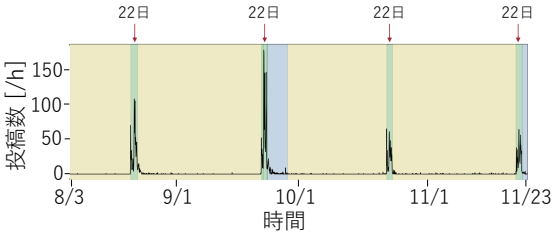


図 7: #ショートケーキの日 を含むツイートの投稿時間

#麻生太郎のデータセット（図 6）では，中央の緑色のセグメントは第 26 回参議院選挙公示日から投票日の 1 週間後までであり，選挙期間を 1 つのセグメントとして捉えている．また，後半 2 つの緑色のセグメントは統一教会関連の報道や沖縄県知事選の期間において沖縄が戦争に巻き込まれる可能性を示唆した発言，国葬問題に関して岸田首相に対し圧をかけた発言が注目され，投稿数が増加している．前半 2 つの黄色のセグメントは選挙に関する投稿，後半の黄色のセグメントは直前の緑のセグメントにおける投稿が収束している期間である．

#ショートケーキの日のデータセット（図 7）では，ショートケーキの日が毎月 22 日であることから，毎月 22 日に投稿が急増するデータセットとなっている．毎月 22 日前後のセグメントを類似レジームで取れていることから，同一パターンのイベント発生を同一レジームで捉えていることがわかる．

6 おわりに

本研究では長期間のイベントデータに対して，複数の Hawkes 過程を用いたイベントデータのセグメンテーションおよびレジーム推定を行うアルゴリズムを提案し，評価検証を行った．Web 上には不定間隔にイベントが発生するデータが多くあることから，Web データのパターン抽出を行うには，不定間隔なデータを扱うことができるアルゴリズムが必要である．提案アルゴリズムは，長期間のデータを意味ある部分系列に分解し，パターン抽出を行うレジーム分類の手法と，イベントデータのモデリングに用いられる点過程の１つである Hawkes 過程の組み合わせで構成されている．イベントデータをより適切にモデリングを行うために Hawkes 過程の対数尤度を用い，モデリングの切り替わり点をイベント変化点としてセグメンテーションを行った．セグメンテーションの有効性を判定するために，最小記述長原理に基づいたコスト関数を設定した．また，セグメンテーションを行うと同時に，レジーム分類を行うことで，イベントデータを意味あるパターンに分類することを実現した．実験では，人工データを用いたレジーム分類によって，提案アルゴリズムの有効性を評価し，また，Web データであるツイートの投稿履歴を用いて，提案アルゴリズムが適切なレジームを推定しているかを検証した．

謝　　辞

本研究の一部は JSPS 科研費，JP20H00585，JP21H03446, JP22K17896, 国立研究開発法人情報通信研究機構委託研究 NICT 03501, 総務省 SCOPE JP192107004，JSTAIP 加速課題 JPMJCR21U4，ERCA 環境研究総合推進費 JPMEERF20201R02, の助成を受けたものです．

文　　献

[1] H. Alvari and P. Shakarian, “Hawkes process for understanding the influence of pathogenic social media accounts,” Proceedings of the The 4th International Conference on Data Intelligence and Security, pp.36–42, 2019.

[2] E. Aramaki, S. Maskawa, and M. Morita, “Twitter catches the flu: detecting influenza epidemics using twitter,” Proceedings of the Conference on empirical methods in natural language processing, pp.1568–1576, 2011.

[3] E. Bacry, I. Mastromatteo, and J.F. Muzy, “Hawkes processes in finance,” Market Microstructure and Liquidity, vol.1, no.01, p.1550005, 2015.

[4] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, “Recurrent marked temporal point processes: Embedding event history to vector,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1555–1564, 2016.

[5] N. Du, M. Farajtabar, A. Ahmed, A.J. Smola, and L. Song, “Dirichlet-hawkes processes with applications to clustering continuous-time document streams,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.219–228, 2015.

[6] H.S. Dutta, V.R. Dutta, A. Adhikary, and T. Chakraborty, “Hawkeseye: Detecting fake retweeters using hawkes process and topic modeling,” IEEE Transactions on Information Forensics and Security, vol.15, pp.2667–2678, 2020.

[7] P.D. Grünwald, I.J. Myung, and M.A. Pitt, Advances in minimum description length: Theory and applications, MIT press, 2005.

[8] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, “Toeplitz inverse covariance-based clustering of multivariate time series data,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.215–223, 2017.

[9] J.A. Hartigan and M.A. Wong, “Algorithm as 136: A k-means clustering algorithm,” Journal of the royal statistical society. series c (applied statistics), vol.28, no.1, pp.100–108, 1979.

[10] A.G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” Biometrika, vol.58, no.1, pp.83–90, 1971.

[11] A.G. Hawkes, “Hawkes processes and their applications to finance: a review,” Quantitative Finance, vol.18, no.2, pp.193–198, 2018.

[12] C. Huang, X. Wu, X. Zhang, C. Zhang, J. Zhao, D. Yin, and N.V. Chawla, “Online purchase prediction via multi-scale modeling of behavior dynamics,” Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining, pp.2613–2622, 2019.

[13] R. Kobayashi and R. Lambiotte, “Tideh: Time-dependent hawkes process for predicting retweet dynamics,” Proceedings of the International AAAI Conference on Web and Social Media, vol.10, no.1, pp.191–200, 2016.

[14] L. Li, B.A. Prakash, and C. Faloutsos, “Parsimonious linear fingerprinting for time series,” Proceedings of the VLDB Endowment, vol.3, no.1-2, pp.385–396, 2010.

[15] Y. Matsubara, Y. Sakurai, and C. Faloutsos, “Autoplait: Automatic mining of co-evolving time sequences,” Proceedings of the ACM SIGMOD International Conference on Management of Data, pp.193–204, 2014.

[16] T. Murayama, S. Wakamiya, E. Aramaki, and R. Kobayashi, “Modeling the spread of fake news on twitter,” Plos one, vol.16, no.4, p.e0250419, 2021.

[17] Y. Ogata, “Statistical models for earthquake occurrences and residual analysis for point processes,” Journal of the American Statistical association, vol.83, no.401, pp.9–27, 1988.

[18] M. Okawa, T. Iwata, Y. Tanaka, H. Toda, T. Kurashima, and H. Kashima, “Dynamic hawkes processes for discovering time-evolving communities’ states behind diffusion processes,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1276–1286, 2021.

[19] M.A. Rizoiu, Y. Lee, S. Mishra, and L. Xie, Hawkes processes for events in social media, Frontiers of multimedia research, 2017.

[20] V. Tozzo, F. Ciech, D. Garbarino, and A. Verri, “Statistical models coupling allows for complex local multivariate time series analysis,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1593-1603, 2021.

[21] Y. Wang, N. Du, R. Trivedi, and L. Song, “Coevolutionary latent feature processes for continuous-time user-item interactions,” Proceedings of the Advances in Neural Information Processing Systems, vol.29, 2016.

[22] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, and J. Leskovec, “Seismic: A self-exciting point process model for predicting tweet popularity,” Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp.1513–1522, 2015.

[23] 近江崇宏, 野村俊一, 点過程の時系列解析, 共立出版, 2019.