DEIM Forum 2022 D21-1(day2 p17)

複合イベントストリームのための多方向特徴自動抽出

中村　航大†,††　松原　靖子†　川畑　光希†　梅田　裕平†††　和田裕一郎†††,††††
櫻井　保志†

† 大阪大学産業科学研究所　〒 567–0047 大阪府茨木市
†† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市
††† 富士通株式会社 人工知能研究所　〒 211–8588 神奈川県川崎市
†††† 理化学研究所 革新知能統合研究センター　〒 103–0027 東京都中央区
E-mail: †{kota88,yasuko,koki,yasushi}@sanken.osaka-u.ac.jp, ††{umeda.yuhei,wada.yuichiro}@fujitsu.com

あらまし　複数の属性情報と時間情報を伴うイベント集合は, オンライン購買履歴 (商品, 価格, ブランド; 購入時刻) やタクシー人口流動データ (乗車地域, 降車地域; 乗車時刻) をはじめとして, 多様な状況下で収集されている. このようなデータは, 大量のイベントデータが複数の属性情報とともに絶え間なく生成される, 複合イベントストリームである. 本論文では複合イベントストリームから, 多方向における特徴を自動で抽出する CompBlast を提案する. CompBlast は (a) 潜在グループと時系列パターンの両方を明らかし, (b) 半無限長となるイベントストリームを簡潔な表現へと自動で要約する. また, (c) 計算時間はデータ長と各属性の次元数に依存せず, 高速に処理を行う. 実データと人工データを用いた実験では, CompBlast が複合イベントストリームから, 潜在グループや時系列パターンといった, 有用な特徴を自動的に発見することを確認した. また, 提案手法が, 最新の既存手法と比較して高精度であり, 計算時間について大幅な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　時系列解析, 複合イベントデータ, テンソル分解, データストリーム処理, 特徴自動抽出

1　ま え が き

位置情報に基づくサービス [1], Web アクティビティ [2], Internet of Things (IoT) サービス [3], 医療情報解析 [4] などの幅広い分野において, 高速かつ大量に時刻付きのイベントデータが生成されている. 例えば, オンライン購買サービスでは, 多数のアイテム情報を伴う, 何百万もの購買イベントが生成される. 主要な需要としては, 潜在的な顧客グループに応じた広告や, 突然の購買行動の変化や不正アクセスといった, 時間発展におけるパターンの変化を検出することである.

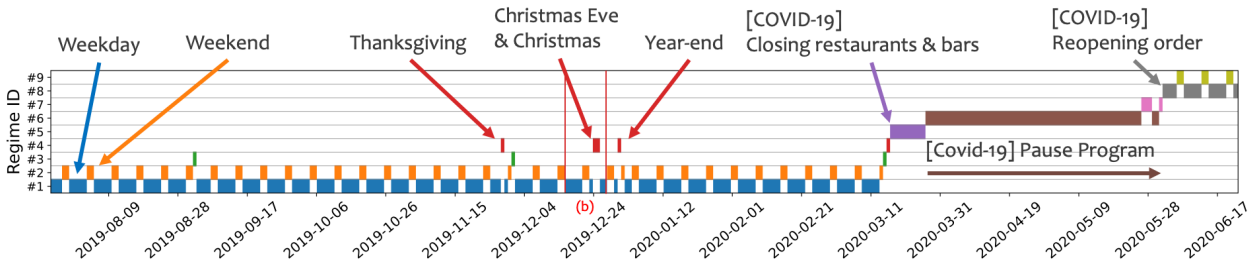
ここで, オンライン購買データ (商品, 価格, ブランド; タイムスタンプ) やタクシー人口流動データ (乗車エリア, 降車エリア; タイムスタンプ) をはじめとした, 複数の属性情報と時間情報によって構成されるイベントデータが絶え間なく生成される状況を “複合イベントストリーム” と定義する. 本研究では, 大規模な複合イベントストリームから重要な特徴を抽出し, 全てのイベント情報を簡潔かつ有用な表現へと要約することを目的とする. 具体的には, 本目的を達成するために以下のような要件を満たす手法が必要である.

- スパースな複合イベントの多方向コンポーネント解析：多数の属性情報が伴うイベント集合から潜在的なグループやトレンドを発見する. 図 1 (d) は (乗車エリア, 降車エリア; タイムスタンプ) で構成される, ニューヨーク市におけるタクシー人口流動データの例である. これは, スパース性を伴う 3 階のテンソルであり, 各属性における潜在グループや周期性といった明確な特徴を全く把握できない. 本研究では, 各属性における潜在グループ/トレンドを示す,“コンポーネント” という新たな概念を導入し, スパースな複合イベントの多方向コンポーネント解析を提案する. この解析結果については, 図 1 (b) (c) の説明とともに後述する.
- イベントストリームにおける自動変化点検出：実社会において, イベントストリームは, 多様かつ異なる期間を持つ時系列パターンを含んでいる. 本研究では, このような時系列パターンを “レジーム” と定義する. 多くの場合, レジーム数や各レジームの期間は事前には未知である. さらに, 複合イベントストリームは, 絶え間なく生成され, 時間発展とともに拡大する半無限長のデータであるため, 高速かつ効率的な処理が求められる. したがって, 自動的かつオンラインに, レジームとそれらの変化点を検出することが必要となる.

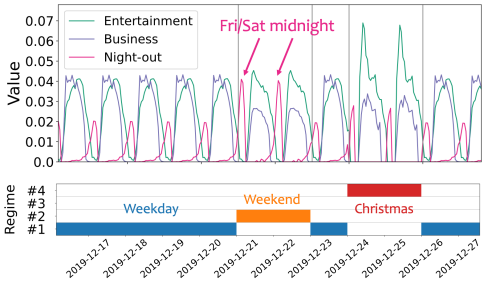
本研究では, 上記の全ての要件を満たし, 複合イベントストリームからの多方向特徴自動抽出を実現するアルゴリズムとして, CompBlast を提案する. 本論文では次の問題を扱う.

問題:. 複数の属性情報と時間情報を持つ, 大規模な複合イベントストリーム $\mathcal{X}$ が与えられたとき, 全てのイベント $\mathcal{X}$ を簡潔かつ効果的な表現へと要約する. より具体的には,

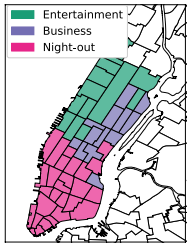
- 各属性における潜在的なグループ/トレンドを示す, コンポーネントを明らかにし,



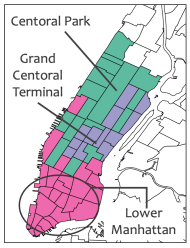
(a) タクシー乗車データにおける, 自動的かつ逐次的なレジーム検出の結果



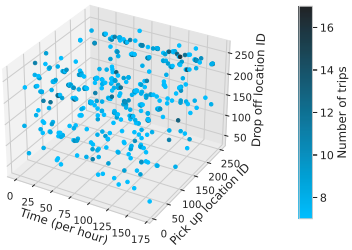
(b) 3 つの代表的なコンポーネントの時系列パターン



(c-i) 乗車エリアの潜在コンポーネント



(c-ii) 降車エリアの潜在コンポーネント



(d) NYC タクシーデータのオリジナルテンソル (1 週間)

図 1　タクシー人口流動データにおける CompBlast の出力例

- 半無限長のデータストリームにおける時系列パターンを示す, レジームとそれらの変化点を検出する.
- また, これらの処理をオンラインかつ自動的に行う.

1.1　具　体　例

図 1 は, 各イベントが 262 種の乗車エリア ID, 264 種の降車エリア ID と乗車時刻の三つの属性によって構成されている, ニューヨーク市のタクシー人口流動データにおける CompBlast のリアルタイム解析の結果を示している.

図 1 (a) に示すように, CompBlast は逐次的かつ自動的に時間的パターン (レジーム) を検出する. より具体的には, まず提案手法は, 平日/休日にそれぞれが一致している, レジーム #1, #2 (青, 橙) を検出した. また, 2019 年の末にかけて, サンクスギビング, クリスマスや年末といった祭日がある. 提案手法は, このような普段とは異なる大衆の動向を, レジーム #3, #4 (緑, 赤) として捉えている. 新たに流行した感染症, COVID-19 は大衆の動向や生活様式やを大きく変えた. ニューヨーク市では, 2020 年 3 月 16 日にレストランやバーは封鎖され, その後から自宅待機プログラムが開始された. CompBlast は, このような大衆の行動変化をレジーム #5 (紫) とレジーム #6 (茶) として検出することに成功している. 続いて, 2020 年 6 月初頭に, 自宅待機プログラムは緩和された. 提案手法は, この緩和による人流の変化を適応的に検出し, 新たな平日と休日の動きを示す, レジーム #8 (灰) とレジーム #9 (暗黄) を生成している. 以上のように, 提案手法は, 適応的かつ自動的に, 複雑な人口流動データにおける, 重要な変化を捉えることができる. さらに重要な点として, 提案手法は, 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, 前述のような潜在的な特徴を捉えることが可能である.

また, CompBlast はコンポーネントとして, 複合イベント集合の潜在トレンドを明らかにする. ここでは, タクシー乗車データにおいて得られたコンポーネントのうち, 主要な 3 つのコンポーネントを, “Entertainment”, “Business”, “Night-out” とラベルを付けた. 図 1 (b) は 2019 年 12 月中旬における, 3 つの主要なコンポーネントの振る舞いと, レジーム割り当てを示す. まず, Entertainment コンポーネントは, 祭日において高い値を持っている. これは, クリスマスなどの祭日におけるニューヨーク市では, 朝から多数のイベントやパレードが開催されるからである. Business コンポーネントは, 休日や祭日と比較して, 平日に高い値を示している. これは, 多くの労働者にとって平日が出勤日であるからだと考えられる. 最後に, Night-out コンポーネントは深夜にピークを示しており, 特に金曜日の夜と土曜日の夜において高い値を示す. これは, 次の日が休日であるこれらの日において, 多くの市民が外出する傾向にあることを示唆している. 以上のように提案手法は, データの理解を助ける, 各コンポーネントの重要な動向を明らかにすることが可能である. 図 1 (c), は, 3 つの主要なコンポーネントと最も関連度の高いエリアを示している. 各コンポーネントの割り当ては, 地理的な特徴を捉えている. 具体的には, Entertainment コンポーネントは, Central Park 周辺や博物館周辺に, Business コンポーネントは, Grand Central Terminal をはじめとした, 主要な駅の周辺に位置している. また, Night-out コンポーネントは, 多数のレストランやバーがある, Lower Manhattan に集中している. ここで重要な点として, 提案手法は, イベントの属性情報に位置関係を示す情報が含まれていないにもかかわらず, コンポーネントと各エリア ID との関係性から, 潜在的な地理的特徴を明らかにしている.

図 1 (d) は, タクシー人口流動データのオリジナルテンソルの例である. 図に示すように, 対象データはスパースなテンソルであり, 各属性に多数の次元を持つ. 提案手法による解析結果とは異なり, レジームやコンポーネントといった明確な特徴を全く把握できない.

1.2 本論文の貢献

本論文では, スパースな複合イベントテンソルストリームからの多方向特徴自動抽出手法として CompBlast を提案する. 提案手法は次の特長を持つ.

（1） 大規模かつスパースな複合イベントにおいて, 潜在トレンドと時系列パターンを捉え, 直感に合致する要約を出力することが可能である.
（2） 時系列パターン, それらの変化点と特徴を事前情報なしに, 自動的に検出することができる.
（3） 増加し続ける高次元複合イベントストリームにおいて, データの全体長と次元数に依存せず, 効率的に処理する.

2 関 連 研 究

コンポーネント解析. Latent Dirichlet Allocation (LDA) [5] やその応用技術 [6] をはじめとした, 混合分布モデルの学習 [7] は大規模離散データ集合の解析を可能にする. 近年では, ニューラルネットワークを用いてモデルの推論を行う, Neural Topic Model (NTM) やその応用技術が提案されている [8]. DISC [9] は最大エントロピー法を用いて, データ内に潜在するグループとそれらの特徴を自動的に抽出する手法を提案した. また, 多数の属性によって構成される離散データ集合はテンソルとして処理することが可能である [10]. TriMine [11] は, 複数の属性情報が伴う, イベント集合のための解析手法である. これらの手法とは異なり, 提案手法は, 属性内における潜在トレンド/グループだけでなく, 時系列パターンも捉えることが可能である. さらに, 特徴抽出はリアルタイムかつ自動で行われる.

レジーム検出. 自己回帰モデル (AR: autoregressive model), 線形動的システム (LDS: linear dynamical model) 隠れマルコフモデル (HMM: hidden Markov model), は代表的な技術であり, これらに基づく時系列解析手法が数多く提案されている [12–14]. TICC [15] は, 有限個の時系列パターンを発見することが可能である. AutoPlait [16] とその発展手法 [17,18] は, 最小記述長 (MDL) 原理を用いて, 自動的に類似時系列パターンを検出する. また, 時系列パターンを用いて高精度の非線形時系列予測を実現する, Orbitmap [19] と CubeCast [20] が提案されている. しかし, 従来の手法は, 値が連続性を持つ時系列シーケンスを対象とするため, スパース性と多数の離散的な属性値を持つイベントテンソルを適切に処理することができない. 深層学習を用いた, 時系列クラスタリングのための手法も提案されている [21]. T-LSTM [22] は不等間隔で得られる, 症状の進行パターンを捉えることが可能である. しかし, これらの手法は高次元テンソルのための手法ではない. また, 計算コストが大きく, Black Box モデルであるためリアルタイム処理においてデータを要約することが困難である.

まとめると, スパースかつ高次元な複合イベントテンソルストリームにおいて, 潜在トレンドと時系列パターンの両方を, リアルタイムに自動抽出する手法は依然として存在しない.

3 提案モデル

本章では, 複合イベントストリームのための解析モデルについて述べる.

3.1 CompBlast の概要

複合イベントストリーム, つまり, 多数の属性情報を伴う, 複合イベントが絶え間なく生成される状況を考える.

[知見 1]（高次元） 複合イベントの集合はテンソルとして扱われる. 図 1 (d) はタクシー人口流動データの複合イベントテンソルである. 各属性には多数の次元を持つ (この図ではそれぞれ, 262, 264)

[知見 2]（スパース） 図に示すように, 各シーケンスは非常にスパースであり, 例えば, $\{0,0,0,1,0,\ 0,1,2,0,\cdots\}$ のように一見すると ただのノイズのようであるため, 典型的な既存の時系列解析手法では適切に処理することができない.

[知見 3]（半無限長） 複合イベントストリームは, 絶え間なく生成され半無限長となるため, データ全体を保持することが困難となる.

まとめると, 本研究の目的は, 上記のような特性を持つ, 複合イベントストリームから簡潔かつ有用なパターンを抽出することである. 具体的には, **(P1)** コンポーネント (潜在トレンド/グループ) と **(P2)** レジーム (時系列パターン) という 2 つの重要なパターンを抽出する.

3.2 問 題 定 義

まず本節では, 提案モデルに必要な概念と問題について定義を行う.

本研究では, それぞれのイベントエントリ (つまり, レコード) が M 個の属性と時間情報で構成される “複合イベント” を扱う. ここで M 個の属性の総数を $U_1 \dots U_M$ とし, データの全長を T とする.

[定義 1]（複合イベントストリーム） $\mathcal{X} \in \mathbb{N}^{U_1 \times \cdots \times U_M \times T}$ を $M+1$ 階のイベントテンソルとする. また, 時間間隔 $\tau \ll T$ で最新時刻 T において, 重複のないカレントテンソル $\mathcal{X}^C \in \mathbb{N}^{U_1 \times \cdots \times U_M \times \tau}$ が与えられる. $\mathcal{X}$ の要素 $x_{u_1 \dots u_M, t}$ は, 各属性における $u_1 \dots u_M$ 次元が, 時刻 t に出現した回数を示す.

(P1) コンポーネントは, 複合イベント集合の主要な振る舞いを表現する. 本手法では, M 個の属性と時間に関するコンポーネントを用いて, 行列 $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(M)}$ と $\mathbf{B}$ として, イベントテンソル $\mathcal{X}^C$ を表現する.

[定義 2]（属性コンポーネント行列 $\mathbf{A}^{(m)}$ $(U_m \times K)$)） 要素 $a^{(m)}_{u,k}$ は m 属性の u 次元と, k 番目の潜在コンポーネントとの関連度の強さを示す. このとき要素 $a^{(m)}_{u,k}$ は正の実数とし, 各要素の合計値を 1 とする ($\sum_{u=1}^{U_m} a^{(m)}_{u,k} = 1$). 同様に, $\mathbf{B}$ は時間コンポーネント行列を表し, $b_{t,k}$ は各時刻とコンポーネントの関連強度を示す.

コンポーネント行列への分解によって, 複合イベント集合を簡潔に表現することが可能であるが, 様々な **(P2)** レジームを

含む複合イベントテンソルストリームのモデリングには不十分である．したがって，より上位の概念を導入することでレジームを表現する．

［定義 3］（レジーム） 特定の類似時系列パターンを表現するために，分解されたコンポーネント行列集合をレジーム θ とする ($\theta = \{\{\mathbf{A}^{(m)}\}_{m=1}^{M}, \mathbf{B}\}$)．$R$ 個のレジームがあるとした時，レジームパラメータ集合として $\Theta = \{\theta_1 \dots \theta_R\}$ を定義する．

また，G 個の時間的パターンの変化点があるとした時，レジーム割り当て集合を $\mathcal{S} = \{s_g\}_{g=1}^{G}$ とする．ここで $s_g = (t_s, r)$ は，時間 t_s に r 番目へ遷移したとしたことを示す．

最終的に，以上の項目を用いて，動的に複合イベントテンソルストリーム $\mathcal{X}$ を表現する．具体的には，$\mathcal{C} = \{R, \Theta, G, \mathcal{S}\}$ を，$\mathcal{X}$ の簡潔な表現とし，候補解と呼ぶ．まとめとして，本論文で扱う問題を次のように定義する．

［問題 1］（リアルタイム要約） カレントテンソル $\mathcal{X}^C$ が与えられたとき，複合イベントテンソルストリーム全体 $\mathcal{X}$ を表現する要約情報 $\mathcal{C}$，すなわち，

- レジーム数 R とレジームパラメータ集合，$\Theta = \{\theta_r\}_{r=1}^{R}$，
- レジーム遷移回数 G とセグメント集合，$\mathcal{S} = \{s_g\}_{g=1}^{G}$．

を逐次的に求めることである．

3.3 CompBlast モデル

本節では，提案モデルの詳細について述べる．CompBlast は以下の二つのアイデアから構成される．

- 多方向コンポーネント分解 (P1): 高次元かつスパースなイベントテンソルからコンポーネント行列 $\{\mathbf{A}^{(m)}\}_{m=1}^{M}, \mathbf{B}$ を抽出する．この分解は，半無限長となる任意次元のイベントテンソルに対応可能である．

- 符号化コスト (P2): レジーム数，レジーム長を決定する，複合イベントストリームのための良い要約を定義する．さらに，最適な要約情報 $\mathcal{C}$ を動的に推定するための手法を提案する．

3.3.1 多方向コンポーネント分解 (P1)

最も単純な場合として，オフライン (テンソルストリーム全体 $\mathcal{X}$ を処理可能な場合) でのモデル化について述べる．第一の課題は複合イベントテンソル $\mathcal{X}$ を表現する，有用な潜在コンポーネント行列 $\{\mathbf{A}^{(m)}\}_{m=1}^{M}, \mathbf{B}$ を抽出することである．そこで本研究では，イベントの生成過程をモデル化する，新たな分解手法を提案する．潜在コンポーネントが現れる過程としては，イベント集合内に潜在する，共通の動機によって発生したイベントが類似した振る舞いを示し，コンポーネントとして現れる．本手法では，トピックモデリングの概念に基づき，上記のイベント生成過程を以下のようにモデル化する．

- For each component $k = 1, \dots, K$:
  - For each attribute $m = 1, \dots, M$:
    - Draw $\mathbf{A}_k^{(m)} \sim \text{Dirichlet}(\alpha^{(m)})$.
- For each time $t = 1, \dots, T$:
  - Draw $\mathbf{B}_t \sim \text{Dirichlet}(\beta)$.
  - For each entry $j = 1, \dots, N_t$:
    - Draw a latent variable $z_{t,j} \sim \text{Multinomial}(\mathbf{B}_t)$.
    - For each attribute $m = 1, \dots, M$:
      - Draw a unit $e_{t,j}^{(m)} \sim \text{Multinomial}(\mathbf{A}_{z_{t,j}}^{(m)})$.

ここで，$\alpha^{(M)}$，β はそれぞれ $\mathbf{A}^M$，$\mathbf{B}$ のためのハイパーパラメータとする．[1]

最終的に，上述の推定を，半無限長となるテンソル全体を保持せずに時間依存性を考慮するモデルへと拡張する．コンポーネントは時事刻々と変化し，それらの平均は，新たなデータ $\mathcal{X}^C$ が観測されない限り，前時刻 $T - \tau$ と同じであると仮定する．この仮定に基づき，以下のディリクレ事前分布を用いることが可能となる: $\text{Dirichlet}(\alpha^{(m)}{}_1\hat{\mathbf{A}}_k^{(m)})$, $\text{Dirichlet}(\beta_1\hat{\mathbf{B}}_t)$, ここで，${}_1\hat{\mathbf{A}}^{(m)}$ と ${}_1\hat{\mathbf{B}}$ は，一時刻前 $T - \tau$ におけるコンポーネント行列である．上記に加えて，より長期間の時間的依存性を導入するために，L 時刻前までのコンポーネント行列に依存するように拡張することが可能である．

$$\text{Draw } \mathbf{A}_k^{(m)} \sim \text{Dirichlet}(\Sigma_{l=1}^{L}\alpha^{(m)}{}_l\hat{\mathbf{A}}_k^{(m)}),$$
$$\text{Draw } \mathbf{B}_t \sim \text{Dirichlet}(\Sigma_{l=1}^{L}\beta_l\hat{\mathbf{B}}_t).$$

3.3.2 符号化コスト (P2)

第二の課題は，レジーム θ として潜在コンポーネント行列 $\{\mathbf{A}^{(m)}\}_{m=1}^{M}, \mathbf{B}$ が与えられたとき，レジーム数とレジーム長を決定し，全てのデータストリームを表現する要約情報 $\mathcal{C}$ を求めることである．そこで本研究では，最小記述長 (Minimum description length: MDL) に基づく，新しい符号化スキームを適用する．直感的には，データがより圧縮できれば，より良いモデルであるとみなす．イベントストリーム全体 $\mathcal{X}$ とそのデータを表現するモデル $\mathcal{C}$ が与えられたとき，総符号化コスト $< \mathcal{X}; \mathcal{C} >$ が最小となるモデルを構築する:

$$< \mathcal{X}; \mathcal{C} > = < \mathcal{C} > + < \mathcal{X} | \mathcal{C} >, \tag{1}$$

ここで，$< \mathcal{C} >$ は “モデル符号化コスト”，$< \mathcal{X} | \mathcal{C} >$ は “データ符号化コスト” である．

モデル符号化コスト. モデル符号化コストは，モデルを記述するために必要とするビット数である．提案モデルではそれぞれ，潜在コンポーネントの次元: $\sum_{m=1}^{M} \log^*(U_m) + \log^*(\tau) + \log^*(K)$ [2]，レジームの個数: $< R >= \log^*(R)$，セグメント数: $< G >= \log^*(G)$，レジーム割り当て: $< \mathcal{S} >= \sum_{s \in \mathcal{S}} < s >$，$< s >= \log^*(t_s) + \log(R)$，レジーム集合: $< \Theta >= \sum_{r=1}^{R} \theta_r$，のコストを必要とする．さらに，レジーム θ の表現コストは次の要素から構成される:

$$< \theta > = \sum_{m=1}^{M} < \mathbf{A}^{(m)} > + < \mathbf{B} >, \tag{2}$$

$$< \mathbf{A}^{(m)} > = |\mathbf{A}^{(m)}| \cdot (\log((U_m - 1) * K) + c^F) + \log^*(|\mathbf{A}^{(m)}|), \tag{3}$$

$$< \mathbf{B} > = |\mathbf{B}| \cdot (\log((K - 1) * \tau) + c^F) + \log^*(|\mathbf{B}|), \tag{4}$$

ここで $|\cdot|$ は，それぞれの行列の非ゼロ要素の総数であり，c^F は浮動小数点のコストである [3]．

データ符号化コスト. ハフマン符号を用いた情報圧縮では，レジーム θ が与えられたときの $\mathcal{X}$ の符号化コストを次のように定義する: $< \mathcal{X} | \theta >= -\log P(\mathcal{X} | \{\mathbf{A}^{(m)}\}_{m=1}^{M}, \mathbf{B})$．したがって，

1：本論文では，$\alpha^{(m)} = \beta = 1/K$ とする．

2：$\log^*$ は整数のユニバーサル符号長を示す．

3：本論文では 8 ビットとする．

候補解 $\mathcal{C}$ が与えられたときの $\mathcal{X}$ の符号化コストは以下である:

$$< \mathcal{X}|\mathcal{C} > = \sum_{r=1}^{R} -\log P(\mathcal{X}[r]|\theta_r), \tag{5}$$

ここで, $\mathcal{X}[r]$ は r 番目のレジームに割り当てられた部分テンソル集合とする.

複合イベントストリームは半無限長のデータであり, ストリーム全体の符号長を計算することは困難であるため, 新たに生成されたデータに対し, 動的に最適な候補解 $\mathcal{C}$ を推定する必要がある. したがって, 最新の部分イベントテンソル $\mathcal{X}^C$ が $\mathcal{X}$ に追加されるときに必要となる総コストの増加量 $\Delta < \mathcal{X}; \mathcal{C} >$ を計算する. より具体的には, レジーム θ_* を用いて, 最新の部分テンソル $\mathcal{X}^C$ を表現するときの増加コスト $\Delta < \mathcal{X}^C; \theta_* >$ が必要となる:

$$< \mathcal{X}^C; \theta_* > = \Delta < \mathcal{C} > + < \mathcal{X}^C|\theta_* >, \tag{6}$$

$$\begin{aligned}\Delta < \mathcal{C} > &= \log^*(R+1) - \log^*(R) + < \theta_* > \\ &\quad + \log^*(G+1) - \log^*(G) + < s > . \end{aligned} \tag{7}$$

$\mathcal{X}^C$ を表現するために, 既存の他のレジームへの遷移を必要とする場合, $\Delta < \mathcal{C} >= \log^*(G+1) - \log^*(G) + < s >$ となる. 新規のレジームを用いて $\mathcal{X}^C$ を表現する場合, 式 7 の全てのコストが必要となる. 上記のどちらでもない場合, $\Delta < \mathcal{C} >= 0$ となる. コストの増加量は, テンソルを表現するために必要なモデルの複雑さと一致する.

4 ストリームアルゴリズム

前章において, 潜在コンポーネントの抽出と符号化スキームにしたがって重要なレジームの探索する, 提案モデルについて説明した. 本章では, このモデル概念を用いて, 効率的に複合イベントストリームの要約を行うためのアルゴリズムである, CompBlast について説明する. CompBlast は 2 つのサブアルゴリズム, B-Decomp と B-Compress によって構成される. 直感的には, 最新のイベントテンソル $\mathcal{X}^C$ からレジーム θ_c を推定し, 推定したレジームを用いて候補解 $\mathcal{C}$ の更新を試みる. 図 2 は, CompBlast の処理の流れを示している. 各サブアルゴリズムについては, 以下の節において説明する.

4.1 B-Decomp

B-Decomp アルゴリズムでは最新の部分テンソル $\mathcal{X}^C$ を表現する, 候補レジーム θ_c としてのコンポーネント行列 $\{\mathbf{A}^{(m)}\}_{m=1}^M$, $\mathbf{B}$ を効率的かつ逐次的に求める. 3.3.1 節の生成過程に従ってコンポーネント行列を推定するために, ギブスサンプリング [23] を用いる. より具体的には, テンソル $\mathcal{X}^C$ 内における非ゼロ要素 $x_{u_1,\ldots,u_M,t}$ に対し, 過去 L ステップ前までの時間的依存性を考慮しながら, 確率 p で潜在変数 $z_{u_1,\ldots,u_M,t}$ を割り振る.

$$\begin{aligned} &p(z_{u_1,\ldots,u_M,t} = k \mid \mathcal{X}^C, \mathbf{B}', \hat{\mathbf{B}}, \beta, \{\mathbf{A}^{(m)\prime}, \hat{\mathbf{A}}^{(m)}, \alpha^{(m)}\}_{m=1}^M) \\ &\propto \frac{b'_{t,k} + \sum_{l=1}^L \beta_l \hat{b}_{t,k}}{\sum_{k=1}^K b'_{t,k} + L\beta} \cdot \prod_{m=1}^M \frac{a^{(m)\prime}_{u_m,k} + \sum_{l=1}^L \alpha^{(m)}{}_l \hat{a}^{(m)}_{u_m,k}}{\sum_{u=1}^{U_m} a^{(m)\prime}_{u,k} + L\alpha^{(m)}}, \end{aligned} \tag{8}$$

ここで, $a^{(m)}_{u_m,k}$ と $b_{t,k}$ は k 番目のコンポーネントに, 各属性の u_m 番目の次元, 時刻 t が割り振られた回数を示す. $b'_{t,k}$ 等のプライム符号は, 時刻 t における, 各属性 m が u_m 番目の次元の

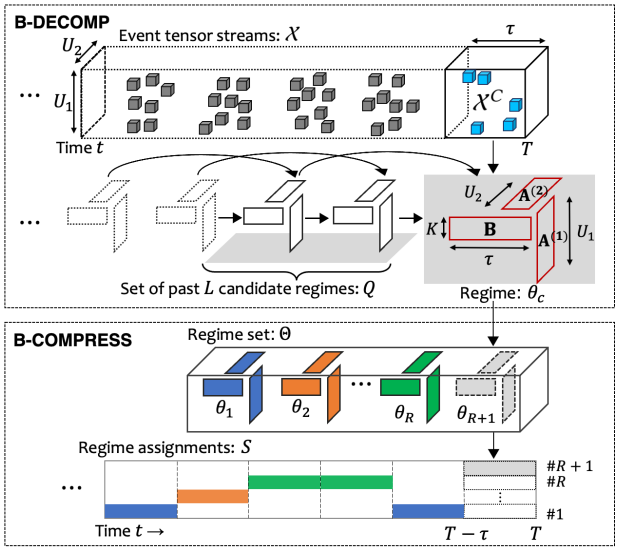


図 2 CompBlast のアルゴリズムの概要

イベントエントリに割り振られた値が除かれていることを示す.

推定される潜在コンポーネント行列 $\{\tilde{\mathbf{A}}^{(m)}\}_{m=1}^M$, $\tilde{\mathbf{B}}$ の各要素は次の式で計算される:

$$\tilde{a}^{(m)}_{u,k} \propto \frac{a^{(m)}_{u,k} + \sum_{l=1}^L \alpha^{(m)}{}_l \hat{a}^{(m)}_{u,k}}{\sum_{u=1}^{U_m} a^{(m)}_{u,k} + L\alpha^{(m)}}, \tilde{b}_{t,k} \propto \frac{b_{t,k} + \sum_{l=1}^L \beta_l \hat{b}_{t,k}}{\sum_{k=1}^K b_{t,k} + L\beta}. \tag{9}$$

重要な点として, この推定は各属性の次元数に対して一定の計算時間で処理を行う (詳細は定理 1).

Algorithm 1 は B-Decomp の詳細を示している. B-Decomp では, 式 (8) によって, テンソル $\mathcal{X}^C$ 内のそれぞれの非ゼロ要素 $x_{u,t}$ に対する潜在変数 $z_{u,t}$ を決定する. 各要素にとっての潜在変数が決定したのち, 式 (9) を用いて潜在コンポーネント行列を推定する. ここで, 過去パラメータはサイズ L の先入れ先出しのキュー Q として扱う. モデルの推定後, キュー Q から最も古いパラメータが取り除かれ, 新たに推定したレジームパラメータが挿入される.

4.2 B-Compress

候補レジーム θ_c を推定後, B-Compress ではレジーム集合 Θ 内のレジームを更新しながら, 候補解 $\mathcal{C}$ を最適化する. Algorithm 2 は B-Compress の詳細を示している. 直前レジーム θ_p と候補レジーム θ_c の二つを監視することで, レジーム遷移を検出する. 式 (6) を用いてそれぞれの増加コストを比較し, コストがより小さくなるように次の手順を決定する.

- 直前レジーム θ_p を採用したときの増加コストが小さい場合, レジーム遷移は発生せず候補レジームは採用されない.
- 候補レジーム θ_c を採用したときの増加コストが小さい場合, 類似レジームの複製を避けるために, Θ の中からより適切なモデルを検索する. その後, コストが最小となるレジームを選択する.

オンラインレジーム更新. 最適なレジームとして既存レジームが選択された場合, 既存レジームは候補レジームに用いて以下の式で更新される.

Algorithm 1 B-Decomp ($\mathcal{X}^C, Q$)

Input: 1. Current tensor $\mathcal{X}^C \in \mathbb{N}^{U_1 \times \ldots \times U_M \times \tau}$
2. Previous model parameter set Q
Output: 1. Current model parameter set $\theta_c = \{\mathbf{A}^{(1)}, \ldots \mathbf{A}^{(M)}, \mathbf{B}\}$
2. Updated previous model parameter set Q'

for each iteration **do**
  for each non-zero element x in $\mathcal{X}^C$ **do**
    for each entry for x **do**
      Draw hidden variable z according to Eq. (8)
    end for
  end for
end for
Compute $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}, \mathbf{B}$ according to Eq. (9)
$\theta_c \leftarrow \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}, \mathbf{B}$;
Q.deque; // Remove oldest previous parameter
$Q' \leftarrow Q$.enque(θ_c); // Insert current model parameter θ_c
return θ_c, Q' ;

$$\tilde{a}^{(m)}_{u,k} \leftarrow \frac{a^{(m)}_{u,k} + \sum_{l=1}^{L} \alpha^{(m)} \hat{a}^{(m)}_{l,u,k} + {}_c a^{(m)}_{u,k}}{\sum_{u=1}^{U_m} a^{(m)}_{u,k} + L\alpha^{(m)} + \sum_{u=1}^{U_m} {}_c a^{(m)}_{u,k}},$$

$$\tilde{b}_{t,k} \leftarrow \frac{b_{t,k} + \sum_{l=1}^{L} \beta \hat{b}_{l,t,k} + {}_c b_{t,k}}{\sum_{k=1}^{K} b_{t,k} + L\beta + \sum_{k=1}^{K} {}_c b_{t,k}}, \qquad (10)$$

ここで, ${}_c a^{(m)}_{u,k}$ 等の符号 c は, 候補レジームの要素を示している. 既存レジームに対する候補レジームの影響度は, 更新するにつれ減衰する. つまり, 各レジームは更新するにつれ収束する.

［定理 1］ 各カレントテンソルにおいて CompBlast は 単位時間あたり最小 $O(N)$, 最大 $O(N+R)$ の計算時間を要する. ここで, N は テンソル $\mathcal{X}$ 内における総イベントエントリ数を示す ($N = \sum_{u,t} x_{u_1,\ldots,u_M,t}$).

［証明 1］ 各時刻において, CompBlast はまず B-Decomp を行う. $\mathcal{X}^C$ 内における各イベントエントリ $x_{u,t}$ において, 潜在変数 $z_{u,t}$ を決定する. コンポーネント数を K, 学習の反復回数を $\#iter$ とすると, この手順は $O(\#iter \cdot KN)$ の計算時間を必要とする. ここで, $\#iter$, K は総イベントエントリ数 N と比較し小さい定数であるため無視することができる. よって, B-Decomp の計算時間は $O(N)$ である. 次に, B-Compress では θ_c と θ_p を監視する. $\mathcal{X}^C$ に適したレジームとして, 前レジーム θ_p が選択された場合, 繰り返し処理を必要とせずに, パラメータが更新されるため計算時間は $O(1)$ のみ必要とする. そうでない場合, レジーム集合 Θ の中から適切なレジームを検索するため, $O(R)$ の計算時間を要する. 全体として, CompBlast はこれらの二つのアルゴリズムによって構成されている. したがって, 単位時間あたり最小 $O(N)$, 最大 $O(N+R)$ の計算時間を要する.

5 評価実験

本論文では, CompBlast の有効性を検証するため, 実データと人工データを用いた実験を行った. 具体的には, 本章では以下の項目について検証を行う.

- 提案手法から得られるレジームとコンポーネントの有効性
- 提案手法のモデリング精度とクラスタリング精度

Algorithm 2 B-Compress ($\theta_c, \mathcal{X}^C, \mathcal{C}$)

Input: 1. Candidate model parameter set $\theta_c = \{\mathbf{A}^{(1)}, \ldots \mathbf{A}^{(M)}, \mathbf{B}\}$
2. New observation tensor $\mathcal{X}^C \in \mathbb{N}^{U_1 \times \ldots \times U_M \times \tau}$
3. Previous candidate solution $\mathcal{C} = \{R, \Theta, G, \mathcal{S}\}$
Output: Updated candidate solution $\mathcal{C}' = \{R', \Theta', G', \mathcal{S}'\}$

if $< \mathcal{X}^C; \theta_p >$ is less than $< \mathcal{X}^C; \theta_c >$ **then**
  /* Stay on the previous regime θ_p */
  $\theta'_p \leftarrow$ Regime update(θ_p, θ_c); according to Eq. (10)
else
  $\theta_e = \arg\min_{\theta \in \Theta} < \mathcal{X}^C; \theta >$;
  if $< \mathcal{X}^C; \theta_c >$ is less than $< \mathcal{X}^C; \theta_e >$ **then**
    /* Shift to the candidate regime θ_c */
    $R' \leftarrow R+1$; $\Theta' \leftarrow \Theta \cup \theta_c$;
    $G' \leftarrow G+1$; $\mathcal{S}' \leftarrow \mathcal{S} \cup (t, R+1)$;
  else
    /* Shift to the existing regime θ_e */
    $\theta'_e \leftarrow$ Regime update(θ_e, θ_c); according to Eq. (10)
    $G' \leftarrow G+1$; $\mathcal{S}' \leftarrow \mathcal{S} \cup (t, e)$;
  end if
end if
return $\mathcal{C}' = \{R', \Theta' G' \mathcal{S}'\}$;

- 複合イベントストリームに対する提案手法の計算コスト

実験には, Intel Xeon E5-2637 3.5GHz quad core CPU, 192GB のメモリを搭載した Linux マシンを使用した. 実験に使用した 3 つのパブリックデータセットは次のとおりである:

- *NY-Taxi*[4]: 2019 年 7 月 1 日から 2020 年 6 月 30 日までの期間におけるニューヨーク市の Yellow Taxi の乗車記録. 各イベントエントリは, 2 つの属性情報 (262 種の乗車エリア ID と 264 種の降車エリア ID) と 1 時間刻みの乗車時間の三つの属性から構成される.
- *Jewelry*[5]: 秘匿化されたオンライン宝石店において収集された, 2 年間にわたる購買記録. それぞれの購買記録は, 価格帯, 6 社の秘匿化されたブランド, 32 種の宝石, 8 種のアクセサリータイプの 4 つの属性情報と 12 時間刻みの購買時刻から構成される. 価格帯は, 最大 $\$1K$ までの価格を \$50 ごとに分類 (離散化) し, それぞれが該当する価格帯を属性情報とした.
- *Electronics*[6]: 様々な電子機器を販売する, 秘匿化されたオンラインストアにおいて収集された, 1 年間の購買履歴. 1 つの取引は, 867 社のブランド, 124 種のアイテムカテゴリと 1 時間ごとの購買時刻によって構成される.

5.1 Q1. 提案手法の有効性

NY-Taxi データセットの結果は 1 章で示したとおりである (図 1). CompBlast は複雑な社会の動向を反映した, レジームとその変化点をオンラインかつ自動で発見した. また, レジーム推移に伴う, コンポーネントの重要な変化と, エリアに潜在す

4：https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
5：https://www.kaggle.com/mkechinov/ecommerce-purchase-history-from-jewelry-store
6：https://www.kaggle.com/mkechinov/ecommerce-purchase-history-from-electronics-store

るグループを明らかにした.

5.2 Q2. 提案手法の精度

続いて, 提案モデルによるモデリング精度とクラスタリング精度について検証する.

モデリング精度. モデリング精度を評価するために, 得られた要約情報を用いてデータを再構成した際の perplexity を評価した. 比較手法は以下の因子分解手法を用いる. (1) Latent Dirichlet Allocation (LDA): 潜在トピック分布として多項分布を用いる, トピックモデリングのための手法. (2) Neural Topic Model (NTM): ニューラルネットワークを用いた変分推定に基づく, トピックモデリングのための手法. 実装と最適化は [8] に従った. (3) TriMine [11]: 複合イベント集合を多項分布を用いてモデリングする, 複数の属性情報が伴うイベント集合のための因子分解手法. 全手法において, データの全期間の半分を訓練期間とし, τ を 1 週間とした. また, 各コンポーネント/トピック数を $4, 8, 16$ と変化させたときの指標を評価した. 図 3 に, 前述の設定における実験での平均 perplexity を示す. perplexity において, 低い値は高いモデル精度を意味する. CompBlast は全てのデータセット, 全てのコンポーネント数において高い精度を示している. 一方, LDA や NTM は, 文章中の単語などの単一の属性情報を効果的に表現することは可能であるが, 多数の属性情報をもつ複合イベント集合を効果的に表現することができず, 提案手法より低い精度を示している. また, TriMine は多数の属性情報をもつイベント集合を表現する能力を持つが, 時間発展するパターン, つまりレジームを捉えることができない.

クラスタリング精度. 続いて, CompBlast の時間方向におけるクラスタリング精度を検証した. 時系列パターンに関して正解ラベルを持つ, 複合イベントストリームを用いてクラスタリング精度の検証を行うために, 人工データを用いる. 人工データの生成プロセスは以下である: 4 つの異なる部分テンソルを生成する (1,2,3,4). 具体的には, それぞれのテンソル $\mathcal{X} \in \mathbb{N}^{100\times100\times100\times100}$ は $100K$ 個のイベントを含む. また, それぞれのテンソルの各イベントエントリは, 乱数 $[0.1, 0.5]$ とディリクレ分布を用いて定義された, 多項分布から生成する. 生成した 4 つのテンソルを異なる組み合わせで時間方向に結合することで, 4 つのデータセットを生成する. 組み合わせは, [15] に従い, “1,2,1”, “1,2,3,2,1”, “1,2,3,4,1,2,3,4”, “1,2,2,1,3,3,3,1” とした. また, CompBlast の比較手法として以下の 3 つのクラスタリング手法を用いる. (1) K-means: ユークリッド距離を用いた, 広く用いられるクラスタリング手法. (2) TICC [15]: 多次元時系列を対象として, 時間方向のセグメンテーションとクラスタリングを同時に行う, オフラインの手法. (3) T-LSTM [22]: 不等間隔で与えられる要素によって構成される系列をクラスタリングする, 時系列クラスタリングのための手法. 図 4 に, 各データセットでの macro-F_1 スコアによるクラスタリング精度を示す. 重要な点として, 比較手法ではクラスタ数を与える必要があるのに対し, 提案手法では, ストリームから自動的にクラスタ数を決定する. したがって本実験では, 比較手法にのみ正解クラスタ数を与える設定で精度を検証している. それにも関

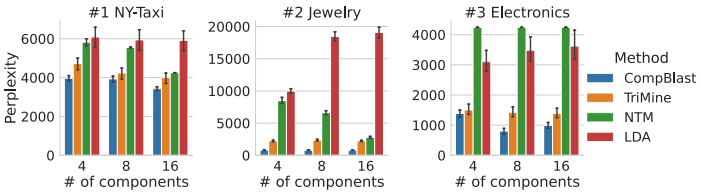


図 3 CompBlast のモデリング精度

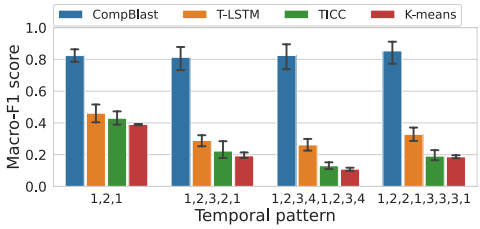


図 4 macro-F_1 スコアによるクラスタリング精度

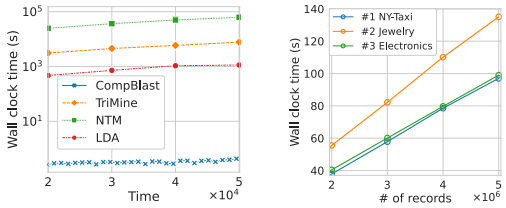


図 5 (左) 各時刻における CompBlast の計算時間
(右) 入力サイズに対する計算コスト

わらず, CompBlast は比較手法に対し, 非常に高い精度を示している. これは, 提案手法のみ, スパースかつ高次元なテンソルを扱うことが可能であるためである. TICC は, 連続的な時系列を対象としているため, スパース性を伴う, テンソルを捉えることができない. T-LSTM は, 不等間隔で与えられるシーケンスを扱うことができるが, 高次元データを処理するための能力を有していない.

5.3 Q3. 提案手法の計算コスト

CompBlast の計算時間に関して, 既存手法と比較することで検証を行った. 図 5 左図は, *NY-Taxi* データセットを用いて処理を行った場合の計算時間である. 提案手法の逐次更新アルゴリズムによって, CompBlast の計算時間はデータ長に依存していない. また, 既存手法よりも高速であり, 最大で 312000 倍の高速化を達成している. 図 5 右図は, 入力テンソルのサイズを変化させたときの計算時間を示す. CompBlast は効率的かつ高速にモデルを推定するため, 全てのデータセットにおいて, 処理するデータ量に対して線形な計算量である. 重要な点として, 計算時間はそれぞれの属性の次元数に依存しない. これは, 各属性において多数の次元を持ちうる, 複合イベントを処理する上で必要不可欠な特性である. 以上のように, 提案手法は, 大規模なイベントストリームの解析に適した性質を持っている.

6 アプリケーション

提案手法によって実現されるアプリケーションの一つである, リアルタイム侵入検出について説明する. 図 6 は *KDDCUP99* データセットにおける, CompBlast のモデリング結果である. 本データセットは, 軍用ネットワーク環境における, 多様なサイバー攻撃をシミュレーションしたものである. 本実験では, 属

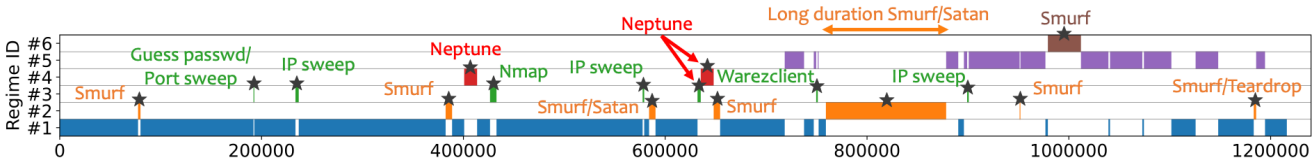


図 6　*KDDCUP99* データセットにおける CompBlast のサイバー攻撃検出とパターン検出

性情報として, 個々の TCP 接続の基本特徴を用いた (つまり, 9 階の複合イベントテンソルストリームを対象とした).

はじめに, CompBlast は, 図中の星印で示された多様なサイバー攻撃に伴い, レジーム遷移を検出している. 時刻 80000 付近において, 長期間の Smurf/Stan 攻撃がある. 提案手法は, 侵入の発生だけでなく, それらの期間も捉えている. 重要な点として, 提案手法は, 教師ラベルや事前学習を必要とせずに, 自動的に異常を検出する. また, 提案手法は, 共通の攻撃を共通のレジームで識別している. 全てのレジーム#2 (橙) は Smurf 攻撃, 全てのレジーム#4 (緑) は Neptune 攻撃と一致している. このような発見は, 未知の攻撃の解釈を助け, 迅速な対策を可能にする. まとめると, CompBlast は有用な解釈性とともに, 多様なサイバー攻撃を検出することが可能である.

7 むすび

本論文では, 大規模かつスパースな複合イベントテンソルストリームのためのリアルタイム解析技術として CompBlast を提案した. 提案手法は, 半無限長となる複合イベントテンソルストリームから, 潜在トレンド/グループや時系列パターンといった, 重要な特徴を自動で検出し, 簡潔な表現へと要約する能力を有する. 評価実験では, 提案手法がレジームやコンポーネントといった潜在的な特徴を自動で発見し, 既存手法と比べて, 高精度なモデリング/クラスタリングを実現することを確認した. また, 計算コストは全データ長や各属性の次元数に依存せず, 高速であることを示した.

謝辞　本研究の一部は JSPS 科研費, JP17H04681, JP18H03245, JP19J11125, JP20H00585, JST さきがけ JPMJPR1659, JST 未来社会創造事業 JPMJMI19B3, JST AIP 加速課題 JPMJCR21U4, 総務省 SCOPE 192107004, 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] Nehme, R. V., Rundensteiner, E. A. and Bertino, E.: Tagging stream data for rich real-time services, *Proceedings of the VLDB Endowment*, Vol. 2, No. 1, pp. 73–84 (2009).
[2] Agarwal, D., Chen, B.-C. and Elango, P.: Spatio-temporal models for estimating click-through rate, *WWW*, pp. 21–30 (2009).
[3] De Francisci Morales, G., Bifet, A., Khan, L., Gama, J. and Fan, W.: Iot big data stream mining, *KDD*, pp. 2119–2120 (2016).
[4] Wang, Y., Chen, R., Ghosh, J., Denny, J. C., Kho, A., Chen, Y., Malin, B. A. and Sun, J.: Rubik: Knowledge guided tensor factorization and completion for health data analytics, *KDD*, pp. 1265–1274 (2015).
[5] Blei, D. M., Ng, A. Y. and Jordan, M. I.: Latent dirichlet allocation, *the Journal of machine Learning research*, Vol. 3, pp. 993–1022 (2003).
[6] Iwata, T., Watanabe, S., Yamada, T. and Ueda, N.: Topic tracking model for analyzing consumer purchase behavior, *IJCAI* (2009).
[7] Okawa, M., Iwata, T., Kurashima, T., Tanaka, Y., Toda, H. and Ueda, N.: Deep Mixture Point Processes: Spatio-temporal Event Prediction with Rich Contextual Information, *KDD*, pp. 373–383 (2019).
[8] Wang, Y., Li, J., Chan, H. P., King, I., Lyu, M. R. and Shi, S.: Topic-Aware Neural Keyphrase Generation for Social Media Language, *ACL*, pp. 2516–2526 (2019).
[9] Dalleiger, S. and Vreeken, J.: Explainable Data Decompositions., *AAAI*, pp. 3709–3716 (2020).
[10] He, H., Henderson, J. and Ho, J. C.: Distributed Tensor Decomposition for Large Scale Health Analytics, *WWW*, ACM, pp. 659–669 (2019).
[11] 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).
[12] Li, L., Prakash, B. A. and Faloutsos, C.: Parsimonious linear fingerprinting for time series, Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE (2010).
[13] Hooi, B., Liu, S., Smailagic, A. and Faloutsos, C.: BeatLex: Summarizing and Forecasting Time Series with Patterns, *PKDD*, Vol. 10535, pp. 3–19 (2017).
[14] Tozzo, V., Ciech, F., Garbarino, D. and Verri, A.: Statistical Models Coupling Allows for Complex Local Multivariate Time Series Analysis, *KDD*, pp. 1593–1603 (2021).
[15] Hallac, D., Vare, S., Boyd, S. and Leskovec, J.: Toeplitz inverse covariance-based clustering of multivariate time series data, *KDD* (2017).
[16] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: AutoPlait: Automatic Mining of Co-evolving Time Sequences, *SIGMOD* (2014).
[17] Kawabata, K., Matsubara, Y. and Sakurai, Y.: Automatic sequential pattern mining in data streams, *CIKM*, pp. 1733–1742 (2019).
[18] Honda, T., Matsubara, Y., Neyama, R., Abe, M. and Sakurai, Y.: Multi-aspect mining of complex sensor sequences, *ICDM* (2019).
[19] Matsubara, Y. and Sakurai, Y.: Dynamic Modeling and Forecasting of Time-Evolving Data Streams, *KDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, New York, NY, USA, Association for Computing Machinery, p. 458–468 (2019).
[20] Kawabata, K., Matsubara, Y., Honda, T. and Sakurai, Y.: Non-Linear Mining of Social Activities in Tensor Streams, *KDD*, pp. 2093–2102 (2020).
[21] Lee, C. and Van Der Schaar, M.: Temporal phenotyping using deep predictive clustering of disease progression, *ICML*, pp. 5767–5777 (2020).
[22] Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K. and Zhou, J.: Patient Subtyping via Time-Aware LSTM Networks, *KDD*, pp. 65–74 (2017).
[23] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. and Welling, M.: Fast collapsed gibbs sampling for latent dirichlet allocation, *KDD*, pp. 569–577 (2008).