情報処理学会論文誌　データベース　Vol.14　No.4　24–35 (Oct. 2021)

複合イベントストリームのための特徴自動抽出

中村 航大[1,2,a)]　松原 靖子[1,b)]　川畑 光希[1,c)]　梅田 裕平[3,d)]　和田 裕一郎[3,4,e)]　櫻井 保志[1,f)]

受付日 2021年3月8日, 採録日 2021年7月2日

概要：複数の属性（乗車時間，乗車エリア，降車エリア，タクシーの種類，顧客の属性…）を含むタクシー乗車データなどに代表される，時間情報をともなうイベント集合は，テンソルストリームとして扱うことができる．本論文では複雑かつ大規模なイベントテンソルストリームから，類似時系列パターンや属性内における潜在グループを自動で抽出する TriComp を提案する．TriComp は (a) 時系列パターンや属性間における類似した特徴を明らかにし，(b) それらの特徴をパラメータのチューニングを行うことなく自動的に抽出し要約する．また，(c) 計算時間はデータストリームの長さに依存せず，高速に処理を行う．実データを用いた実験では，TriComp が複雑なイベントストリームから時系列変化を正確にとらえ，潜在グループや時系列パターンといった，データの解釈を助ける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

Automatic Mining of Complex Event Streams

Kota Nakamura[1,2,a)]　Yasuko Matsubara[1,b)]　Koki Kawabata[1,c)]
Yuhei Umeda[3,d)]　Yuichiro Wada[3,4,e)]　Yasushi Sakurai[1,f)]

Received: March 8, 2021, Accepted: July 2, 2021

Abstract: Given that large tensor streams of time-evolving events such as taxi rides, which contain multiple attributes (e.g., pick up time, pick up area, drop off area, taxi type, customer attribute…) are difficult to comprehend, how do we obtain intuitive groups and patterns? Also, how do we incrementally capture latent structure and typical patterns to achieve a meaningful summarization? In this paper, we propose a streaming algorithm, namely TriComp, which is designed to automatically find both typical patterns and latent groups in such complex yet huge collections. Our method has the following advantages: (a) *it is Effective*: it provides compact and powerful representations that reveal similar features with respect to both time and attributes. (b) *it is Automatic*: it automatically recognizes and summarizes them without any parameter tuning. (c) *it is Scalable*: it is incremental yet scalable, and thus requires computational time that is independent of data stream length. Extensive experiments on real datasets demonstrate that TriComp provides a summarization that helps us understand the complicated data and that consistently outperforms the state-of-the-art methods in terms of both execution speed and accuracy.

Keywords: time series analysis, complex events, tensor decomposition, stream processing, automatic mining

[1] 大阪大学産業科学研究所
SANKEN Osaka University, Ibaraki, Osaka 567–0047, Japan
[2] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
[3] 株式会社富士通研究所
FUJITSU Laboratories LTD., Kawasaki, Kanagawa 211–8588, Japan
[4] 理化学研究所革新知能統合研究センター
RIKEN AIP, Chuo, Tokyo 103–0027, Japan

[a)] kota88@sanken.osaka-u.ac.jp
[b)] yasuko@sanken.osaka-u.ac.jp
[c)] koki@sanken.osaka-u.ac.jp
[d)] umeda.yuhei@fujitsu.com
[e)] wada.yuichiro@fujitsu.com
[f)] yasushi@sanken.osaka-u.ac.jp

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]，医療情報解析 [3], [4]，などの幅広い分野において，複数の属性情報を持つイベントデータが毎時刻生成されている．このようなデータは，複雑でありながら大量に生成され続けるため，リアルタイムに有用な要約を獲得することが重要である．具体的には，潜在的なグループを明らかにするような解釈性，実社会の状況に合わせて変化する適応性，人手を介さない自動性をあわせ持つ要約が必要とされる．たとえば，タクシー乗車データは，乗車エリア，降車エリア，顧客情報，タクシーの種類といった情報が付加されている．このようなデータを，稼働状況を反映しながら効果的なマーケティングに活用するために，エリアや顧客情報にどのような潜在的なグループがあるのか，夏季と冬季でどのような違いがあるのか，といった分析をチューニングによる時間的コストや人材コストを必要とせずに行うことが求められる．したがって本研究では，大量のイベントデータから要約されるべき情報を時系列パターンと潜在グループの集合と定義し，要約情報と呼ぶ．

上述の需要を満たす有用な要約情報を獲得するために，以下の 2 つの重要かつ困難な課題の解決が必要となる．

- 非構造データの構造化：潜在的に存在するグループとそのグループに対する各属性の関連度の強さを明らかにする．多数の属性を含むイベントデータストリームはセンサデータのような連続的な系列データとは異なり，スパースで大規模なテンソルとして表現されるため扱いが困難である．本論文では，このような複数の属性を持つログデータを「複合イベント」と定義する．
- 時系列パターンの発見：データストリームに現れる類似時系列パターンを発見する．それぞれの類似時系列パターンは多様な種類と異なるパターン長を持つため，種類数，特徴，パターン長をデータから自動的に学習することが必要となる．本論文では，このような類似時系列パターンを「レジーム」と定義する．

本研究では，リアルタイム処理において，大量に発生する複合イベント集合から，潜在グループとレジームの両方を自動で発見する手法として TriComp を提案する．具体的には，それぞれのイベントデータが時刻（time）と 2 つの属性情報（entity1，entity2）を持つとして，以下の課題に取り組む．

問題 1　3 つ組 (entity1, entity2, time) で構成されるイベント群が与えられたとき，

- 潜在的に存在するグループとそれらのグループに対する各属性の関連度の強さを明らかにし，
- 自動ですべての時系列パターンを抽出し，潜在グループと時系列パターンの両方をモデルとして表現する．
- また，これらの処理をオンラインかつ高速に行う．

なお，提案手法は上述の 3 つ組以外にも任意の属性数を持つイベントを扱うことができるが，論述の簡略化のために本論文では主に 3 つ組のイベントについてのみ言及する．

1.1 具体例

図 1 は本研究で対象とする複合イベントテンソルの例である．図中の各データ点は，乗車エリア，降車エリア，乗車時間の 3 つ組で構成されるイベントに対応する．オリジナルデータはスパースなテンソルであり，時系列パターンやグループを発見できず，明確な特徴をまったく把握できない．

このような複雑なイベントテンソルストリームにおいて TriComp は類似時系列パターン（レジーム）を自動的に発見する．**図 2** (a) は TriComp の出力結果である．提案手法は，はじめにレジーム 1 として時系列パターンと潜在グループをモデル化した．時刻 80 では，パターン変化を自動的に検出し，レジーム 2 を新たに生成することで異なる特徴を持つパターンを表現している．最終的に得られるレジームの変化点と割当てから，レジーム 1 が平日，レジーム 2 が休日に対応していることが分かる．また，図 2 (a) の赤枠は提案手法がレジーム 2 に割り当てた期間であるが，この日は実際に祝日であることから，周期性のない特徴も把握することに成功している．結果として，実際の社会活動と一致するような特徴を持つ，複数のレジームを自動で検出している．このように，提案手法はデータに関する事前情報を必要とせず，イベントテンソルストリームから有用なパターンを検出する．

また，TriComp はイベントテンソルストリームに潜在的に存在する共通のグループとそれらへの各属性の関連度を要約情報として抽出する．図 2 (a) において，各シーケンスは，それぞれのグループに対する各時刻の関連度の強さを示す．図 2 (b)，(c) では，提案手法が検出した 3 つの潜在グループが存在し，それらに対する各エリアの関連度は色の濃さで示されている．

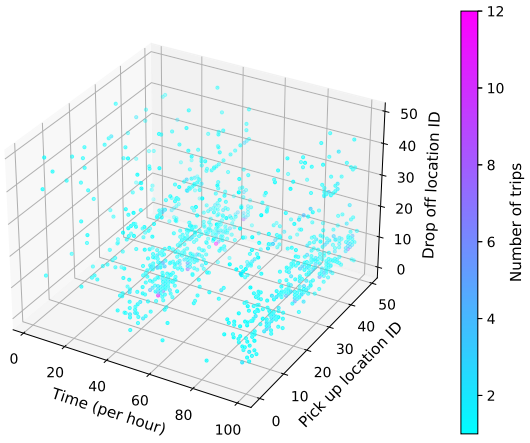
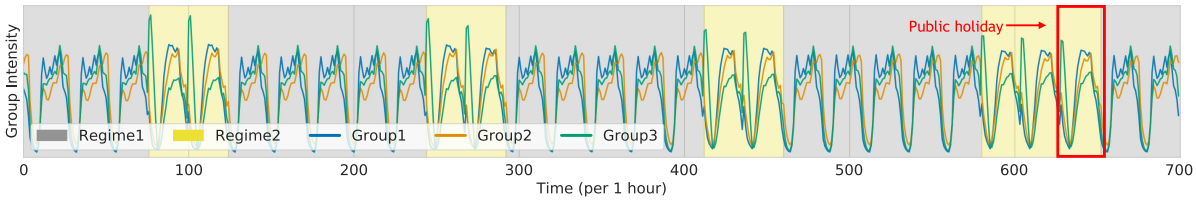
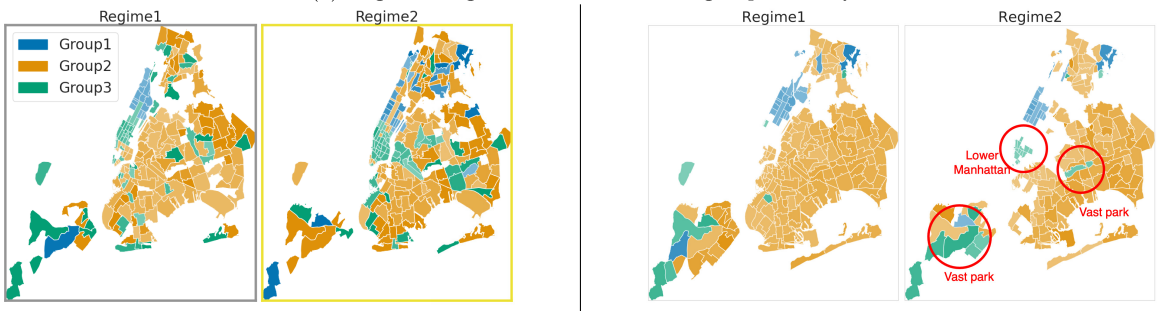


図 1　オリジナルのイベントテンソルストリームの一部
Fig. 1　Part of original event tensor stream.



(a) Regime assignments and time-wise group intensity



(b) Pick up location-specific groups　　(c) Drop off location-specific groups

図 2　(a) レジーム割当てと時間ごとのグループ関連度 (b)，(c) 3 つの潜在グループとそれらに対する各エリアの関連度（色の濃さ）．0.4 以下の小さな関連度は可視化していない．

Fig. 2　Modeling power of TriComp for taxi ride events: (a) Segments (shaded rectangle with yellow and gray) and time-wise intensity sequences of each group. (b), (c) Each attribute (pick up and drop off area)-specific groups. It shows latent groups (colors) and its attributes wise participation weights (depth of colors). Too low degrees are not shown (< 0.4).

具体的な例としてレジーム 2（休日レジーム）を取り上げる．図 2 (a) において，レジーム 2 ではグループ 3（緑色）が強い関連度を示しており，グループ 3 は休日との関連が強いグループであることが分かる．また，休日においてグループ 3 へと変化した赤円で囲まれたエリアが，広大な自然公園や，多くのバーやレストランがある Lower Manhattan であることを考慮すると，グループ 3 は娯楽に関連するグループであると考えられる．また，グループは属性間で共通であるため，娯楽グループの乗車は図 2 (b) の右図において緑色のエリア（グループ 3）から多く発生していることが分かる．以上のように，TriComp は，事前知識を要することなく明確な特徴をとらえ，データの解釈を助ける，有益な要約情報を抽出することができる．

1.2　本論文の貢献

本論文では複合イベントテンソルストリームにおける特徴自動抽出手法として TriComp を提案する．提案手法は次の特長を持つ．

(1) 大規模かつ複雑な複合イベントから，有用な特徴である時系列パターンと潜在グループの両方を動的に抽出する．
(2) 上記の特徴抽出は自動的に行われ，ユーザの介入を必要とせずに発見することができる．
(3) 増加し続ける複合イベントストリームにおいて，データ全体を保持せず効率的に処理することができる．

2.　関連研究

行列/テンソル分解． 行列分解に基づく手法は高次元データが持つ潜在的に存在する要素の発見に有用である．CP 分解 [5] や Tucker 分解 [6] は基礎的な技術であり，幅広い分野で用いられている．さらに，時間情報を含むデータに対してより効果的な手法として，時間発展を考慮した分解手法が数多く提案されている [7], [8]．CompCube [9] や PowerCast [10] は非線形の時間発展をとらえ，効果的な時系列解析を行う．RobustSTL [11]，Fast RobustSTL [12] は，季節性の変動と変化点をとらえることが可能である．しかし，これらの連続値を対象とした手法は，離散値で構成される非構造な複合イベントを適切に扱うことができない．

複数の属性を持つ離散データはテンソルとして処理することが可能である [13]．Rubik [4] は，遺伝子に関する専門知識をモデルに組み込むことで，表現型[*1]を明らかにする，スパース性を持つテンソルのための手法である．また，確率分布を用いた分解も潜在的な構造を発見するうえで有用である [14], [15], [16]．TriMine [17] はトピックモデルに基づく，拡張性を兼ね備えた手法である．Dalleiger らは，最大エントロピー法を用いて，データ内のグループとそれらの特徴を発見する手法を提案した [18]．深層学習を用いた

[*1] 生物の示す形態的，生理的な性質．遺伝子に規定されて発現する形質．

手法も多数提案されている [19], [20], [21]．CoSTCo [22] はスパーステンソルのための畳み込みニューラルネットワークに基づくモデルである．これらの手法とは異なり，提案手法は，時系列パターンと属性内における潜在グループの両方を要約情報として抽出する．さらに，特徴抽出はリアルタイムかつ自動で行われる．

動的モデリングに基づく要約． 隠れマルコフモデル（HMM: hidden Markov model），自己回帰モデル（AR: autoregressive model），線形動的システム（LDS: linear dynamical 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 [23], [24], [25]．AutoPlait [26] と TICC [27] は，センサデータなどの多次元時系列データから類似した時系列パターンを発見する手法である．StreamScope [28] は AutoPlait を発展させた，リアルタイムに処理を行うモデルであり，CubeMarker [29] は 3 階のテンソルからパターンを検出することが可能である．また，CubeCast [30] はテンソルの要約情報を利用し，効果的な非線形予測を行う．一方，連続的な時系列シーケンスを対象とした従来の手法と異なり，提案手法は，スパース性がともなうイベントテンソルのために設計されている．

まとめると，大規模イベントテンソルストリームにおいて，潜在グループと類似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．本研究の目的は，この問題を解決するため，イベントテンソルストリームのための有用なテンソル分解と類似時系列パターンの検出に基づく，ストリーム処理指向の自動解析モデルを開発することである．

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．提案モデルに必要な概念と問題について定義を行ったのち，それらの詳細について説明する．

3.1 問題定義

表 1 に本研究で使用する記号の定義を示す．本研究では，2 つの属性と時間情報 (entity1, entity2, time) の 3 つ組で構成される複合イベントを扱う．ここで entity1，entity2 の総数をそれぞれ u，v とし，総タイムスタンプ数を n とする．

定義 1（イベントテンソル） $\mathcal{X} \in \mathbb{N}^{u \times v \times n}$ を 3 階のイベントテンソルとする．$\mathcal{X}$ の要素 $x_{i,j,t}$ は時刻 t において entity1 の i 番目に entity2 の j 番目が出現した頻度を示す．

また本論文では，各イベントエントリに共通の潜在グループが存在すると仮定する．これにより，TriComp は (entity1, entity2, time) の 3 要素に対し潜在グループを発見し，テンソル $\mathcal{X}$ を 3 つの行列（$\mathbf{A}$，$\mathbf{B}$，$\mathbf{C}$）に分解する．

定義 2（潜在行列 $\mathbf{A}$ $(u \times k)$） 要素 $a_{i,j}$ は i 番目の entity1 と j 番目の潜在グループとの関連度の強さを示す．

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

記号	定義
u，v	entity1 と entity2 の総数
n	イベントテンソルの長さ
$\mathcal{X}$	3 階イベントテンソル $\mathcal{X} \in \mathbb{N}^{u \times v \times n}$
k	潜在グループの個数
$\mathbf{A}$，$\hat{\mathbf{A}}$	entity1 に関する潜在行列と過去のモデルパラメータ，$u \times k$
$\mathbf{B}$，$\hat{\mathbf{B}}$	entity2 に関する潜在行列と過去のモデルパラメータ，$k \times v$
$\mathbf{C}$，$\hat{\mathbf{C}}$	時間に関する潜在行列と過去のモデルパラメータ，$k \times n$
θ_i	i 番目のモデルパラメータ集合 $\theta_i = \{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i\}$
Θ	q 個のレジームによる全レジーム集合，$\Theta = \{\theta_1 \dots \theta_q\}$
s_i	i 番目のレジームの遷移履歴
S	q 個のレジームによる遷移履歴集合 $S = \{s_1 \dots s_q\}$
$\mathbf{D}$	候補解　$\mathbf{D} = \{\Theta, S\}$

このとき要素 $a_{i,j}$ は正の実数とし，各要素の合計値を 1 とする（$\sum_j a_{i,j} = 1$）．entity2 に関する潜在行列 $\mathbf{B}$ と時間に関する潜在行列 $\mathbf{C}$ の定義も上記と同様であるが，簡略化のため省略する．潜在行列 $\mathbf{A}$，$\mathbf{B}$，$\mathbf{C}$ はそれぞれ，(entity1, entity2, time) の各要素における潜在グループ#1, #2, ..., #k に対する関連度の強さを示す．

提案手法は 3 つ以上の属性（$M > 3$）を持つイベントを扱うことも可能である．複合イベント集合が M 階テンソル $\mathcal{X}$ で表現されるとき，提案手法は，$\mathcal{X}$ を M 個の潜在行列に分解することができる（$\mathbf{A}, \mathbf{B}^{(1)}, \dots, \mathbf{B}^{(M-2)}, \mathbf{C}$）．

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（複合イベント集合からの潜在グループの発見） 3 つ組 (entity1，entity2，time) で構成されるイベントテンソル $\mathcal{X}$ が与えられたとき，$\mathcal{X}$ の潜在グループを明らかにし，(entity1, entity2, time) の各要素に対し潜在行列を得る．

上述のように，M 個の行列への分解によって，複合イベントから潜在グループを発見をすることが可能であるが，様々な時系列パターンを含む複合イベントテンソルストリームの表現には不十分である．したがって，時間発展にともなう，潜在グループとそれらへの関連度の変化を表現する必要がある．

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 3 つの潜在行列をレジーム θ とする（$\theta = \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$）．$q$ 個のレジームがあると仮定したとき，レジームパラメータ集合として $\Theta = \{\theta_1 \dots \theta_q\}$ を定義する．また，レジーム割当て集合を $S = \{s_1 \dots s_q\}$ とする．ここで $s_i = \{(t_s, j), \dots\}$ は，時間 t_s に i 番目のレジームから j 番目のレジームに遷移したことを示す．

定義 4（候補解） $\mathbf{D} = \{\Theta, S\}$ を $\mathcal{X}$ を表現する全パラメータ集合とし，候補解と呼ぶ．

本論文の目的は，複合イベントストリーム $\mathcal{X}$ に潜在するグループを抽出すると同時に，類似時系列パターンを発見し，$\mathcal{X}$ の有用な要約を行うことである．本論文で扱う問題を以下のように定義する．

問題 2（リアルタイム要約） 複合イベントテンソルストリーム $\mathcal{X}$ が与えられたとき，$\mathcal{X}$ 全体を表現する要約情報 $\mathbf{D}$，すなわち，

- レジームの個数 q,
- レジームパラメータ集合，$\Theta = \{\theta_1 \dots \theta_q\}$,
- それらの割当て集合，$S = \{s_1 \dots s_q\}$.

を求めることである．

3.2 TriComp モデル

本節では，提案モデルの詳細について述べる．TriComp は以下の 2 つのアイデアから構成される．

- オンラインテンソル分解：複雑かつ大規模なイベントテンソルから潜在グループの抽出を可能にする多方向分解を行う．さらに重要な点として，この分解は逐次的かつ高速に処理を行う．
- 自動圧縮：新しい符号化スキームを導入することで，イベントテンソルストリームから抽出された要約情報を自動的に評価する．この評価に基づいて，時系列パターンを検出し，潜在グループと時系列パターンの両方をモデルとして表現する．

3.2.1 オンラインテンソル分解

第 1 の課題はスパース性をともなう複合イベントストリームから潜在グループを抽出することである．まず，オフライン（単一のイベントテンソル $\mathcal{X}$ のみ扱う場合）での分解について述べる．M 階の複合イベントテンソルが与えられたとき，k 個のグループを発見し，これらのグループに基づく M 個の潜在行列を推定する．本手法ではそれぞれのイベントエントリに対し 1 つの潜在グループを割り当てる．イベント集合における生成モデルは以下のとおりである．

(1) For each groups $r = 1, \dots, k$:
  (a) For each tensor mode $m = 1, \dots, M-2$:
    (i) Draw $\mathbf{B}_r^{(m)} \sim \text{Dirichlet}(\beta^{(m)})$.
  (b) Draw $\mathbf{C}_r \sim \text{Dirichlet}(\gamma)$.
(2) For each entity1 $i = 1, \dots, u$:
  (a) Draw $\mathbf{A}_i \sim \text{Dirichlet}(\alpha)$.
  (b) For each entry $j = 1, \dots, N_i$:
    (i) Draw a latent variable $z_{i,j} \sim$ Multinomial($\mathbf{A}_i$).
    (ii) For each tensor mode $m = 1, \dots, M-2$:
      (A) Draw an entity2 $e_{i,j}^{(m)} \sim$ Multinomial($\mathbf{B}_{z_{i,j}}^{(m)}$).
    (iii) Draw a timestamp $t_{i,j} \sim$ Multinomial($\mathbf{C}_{z_{i,j}}$).

ここで，α，$\beta^{(m)}$，γ はそれぞれ $\mathbf{A}$，$\mathbf{B}^{(m)}$，$\mathbf{C}$ のための固定パラメータとする[*2]．

次に，上述の推定をオンラインで効率的に行う手法について述べる．提案手法では，以前のモデルパラメータを利用することで，過去のモデルの情報を引き継ぐことを可能にする．(entity1, entity2, time) の各要素における各潜在グループへの関連度は，時々刻々と変化し，現時刻におけるそれらの関連度は，新たなデータが観測されない限り一時刻前の関連度と同じであると仮定する．具体的には，それぞれの潜在行列における過去のグループ関連強度 $\hat{\mathbf{A}}_{t-1,u}$，$\hat{\mathbf{B}}_{t-1,k}^{(m)}$，$\hat{\mathbf{C}}_{t-1,k}$ を，それぞれのディリクレ事前分布の平均としてパラメータに組み込む（Dirichlet($\alpha\hat{a}_{t-1,i}$)，Dirichlet($\beta^{(m)}\hat{b}_{t-1,r}^{(m)}$)，Dirichlet($\gamma\hat{c}_{t-1,r}$)）．

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 ステップ前までの時系列変化を考慮するとき，ディリクレ事前分布は以下のように表される．

$$\begin{aligned} &Draw\ \mathbf{A}_i \sim Dirichlet(\Sigma_{l=1}^{L}\alpha\hat{a}_{t-l,i}),\\ &Draw\ \mathbf{B}_r^{(m)} \sim Dirichlet(\Sigma_{l=1}^{L}\beta^{(m)}\hat{b}_{t-l,r}^{(m)}),\\ &Draw\ \mathbf{C}_r \sim Dirichlet(\Sigma_{l=1}^{L}\gamma\hat{c}_{t-l,r}). \end{aligned} \tag{1}$$

過去パラメータの導入によって，時系列変化をモデル化するために過去のテンソルを保持する必要がなくなり，省計算時間かつ省メモリ容量で処理を行うことが可能となる．

3.2.2 自動圧縮

第 2 の課題は，潜在行列 $\mathbf{A}$，$\mathbf{B}^{(m)}$，$\mathbf{C}$ が与えられたとき，複合イベントテンソルストリームを表現する良い要約情報を定義し，自動でモデルを構築することである．本研究では，良い要約情報を定義するため，最小記述長（Minimum description length: MDL）に基づく符号化スキームを適用する．MDL に従い，候補解 $\mathbf{D}$ を表現するための「モデル表現コスト」，候補解 $\mathbf{D}$ が与えられたときのデータ $\mathcal{X}$ の「符号化コスト」を定義し，これらの総和が最小となるモデルを構築する．

モデル表現コスト． 本研究におけるモデル表現コストはすべてのレジームを表現するためのコスト $<\Theta>$ によって定義される．浮動小数点のコストを c_F とすると[*3]，$<\Theta>$ は次の要素から構成される[*4]：

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

$$<\mathbf{A}> = |\mathbf{A}| \cdot (\log((k-1)*u) + c_F) + \log^*(|\mathbf{A}|), \tag{3}$$

$$<\mathbf{B}> = |\mathbf{B}| \cdot (\log((v-1)*k)) + c_F) + \log^*(|\mathbf{B}|), \tag{4}$$

$$<\mathbf{C}> = |\mathbf{C}| \cdot (\log((n-1)*k) + c_F) + \log^*(|\mathbf{C}|). \tag{5}$$

ここで $|\cdot|$ は，それぞれの行列の要素と $1/k$，$1/v$，$1/n$ との差における，非ゼロ要素の総数である．

[*2] 本論文では，$\alpha = \frac{0.5}{k}$，$\beta^{(m)} = 0.1$，$\gamma = 0.1$ とする．
[*3] 本論文では 8 ビットとする．
[*4] $\log^*$ は整数のユニバーサル符号長を示す．

データの符号化コスト．ハフマン符号を用いた情報圧縮では，候補解 $\mathbf{D}$ 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：

$$\begin{aligned} <\mathcal{X}|\mathbf{D}> &= \sum_{p=1}^{q} <\mathcal{X}[s_p]|\Theta> \\ &= \sum_{p=1}^{q} -\log P(\mathcal{X}[s_p]|\theta_p). \end{aligned} \tag{6}$$

ここで，$\mathcal{X}[s_p]$ は p 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，データ全体の符号長を計算することは困難である．したがって，新たに生成されたデータに対して動的にパラメータを最適化する．より具体的には，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を計算し，増加量が最小となるようにレジームパラメータ集合 Θ とそれらの割当て S を求める．$\mathcal{X}^C$ を表現するために追加で必要となるコストは以下のようになる：

$$\begin{aligned} \Delta<\mathcal{X};\mathbf{D}> &= <\mathcal{X}^C;\theta_*> \\ &= \Delta<\theta_*> + <\mathcal{X}^C|\theta_*>. \end{aligned} \tag{7}$$

ここで，θ_* は $\mathcal{X}^C$ を表現するために用いるレジームパラメータである．$\mathcal{X}^C$ を既存レジームによって表現できる場合は $\Delta<\theta_*> = 0$ となり，そうでない場合，新しいレジームを記述するためのモデル表現コストが必要となる．上述の計算における新しいレジームの採用には，追加のモデル表現コストが必要となるため，既存のレジームより高い表現力（低い符号化コスト）でデータを表現することが求められる．したがって，総コストの増加量が最小となるようにモデルを構築することで，データを表現するうえで冗長なレジームを含まない，簡潔かつ効果的なレジームパラメータ集合 Θ の構築を行うことができる．

4. ストリームアルゴリズム

本章では複合イベントテンソルストリームを，高速かつ自動で解析するためのアルゴリズムである TriComp について述べる．

4.1 概要

前章で述べた符号化理論に従い，有用な要約情報を抽出するためには (a) 複合イベントのテンソル分解に基づいたモデルの推定，(b) 候補解 $\mathbf{D}$ のリアルタイム最適化を行う必要がある．これらを達成するためのアルゴリズムである TriComp の概要を Algorithm 1 に示す．また，**図 3** は，TriComp の処理の流れを示している．直感的には，最新のイベントテンソル $\mathcal{X}^C$ からモデルパラメータ集合（レジーム）を推定し，推定レジームを用いて候補解 $\mathbf{D}$ の更新を試みる．より具体的には，アルゴリズムは以下の 2 つの手順で構成されている．

Algorithm 1 TriComp $(\mathcal{X}^C, \mathbf{D})$

Input: 1.Current tensor $\mathcal{X}^C \in \mathbb{N}^{u\times v\times \tau}$
2.Previous candidate solution $\mathbf{D} = \{\Theta, S\}$
Output: Updated candidate solution $\mathbf{D}'$
1: θ = TriComp-deComp $(\mathcal{X}^C)$;
2: Θ', S' = TriComp-Compress $(\mathcal{X}^C, \mathbf{D}, \theta)$;
3: **return** $\mathbf{D}' = \{\Theta', S'\}$;

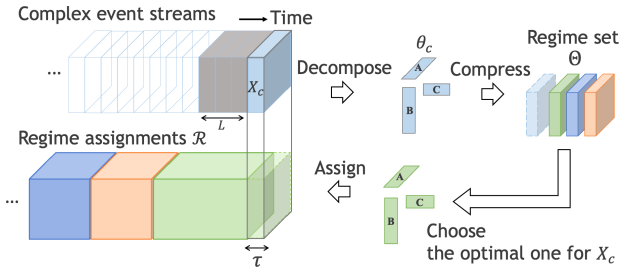


図 3　TriComp のアルゴリズムの概要
Fig. 3　An overview of TriComp.

(P1)　TriComp-deComp：部分イベントテンソルから候補レジーム θ_c を推定する．θ_c は L ステップ前までの時間的依存性を考慮した，オンラインテンソル分解によって導出される．

(P2)　TriComp-Compress：直前レジーム θ_p と候補レジーム θ_c を監視しながら，評価指標である式 (7) に基づいて最適なレジームを採用する．また，レジーム集合 Θ は，モデルの切替えをともなう，提案手法に適した手順で更新される．

ここで，$\mathcal{X}$ の部分テンソルとしての $\mathcal{X}^C$ を，データ長が $\tau \ll n$ であり，要素集合 $x_{i,j,t-\tau+1}, \ldots, x_{i,j,t}$ を持つとする．本アルゴリズムでは，重複のない $\mathcal{X}^C$ が，τ 間隔で与えられるとする．以降の説明では，論文の簡略化のためにイベントテンソルを 3 階のテンソルとして述べるが，潜在行列 $\mathbf{B}$ に関して拡張するだけで，より高次元のテンソルを扱うことが可能である．

4.2 TriComp-deComp

本研究では，ギブスサンプリング [31] を用いて潜在グループの推定を行う．テンソル $\mathcal{X}^C$ 内における非ゼロ要素 $x_{i,j,t}$ に対し，確率 p で潜在グループを割り振る．それぞれの要素にとっての潜在グループ $z_{i,j,t}$ は，過去パラメータを考慮しながら以下の確率によって決定される．

$$\begin{aligned} &p(z_{i,j,t} = r|\mathcal{X}, \mathbf{A}', \mathbf{B}', \mathbf{C}', \alpha, \beta, \gamma, \hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}) \\ &\propto \frac{a'_{i,r} + \sum_{l=1}^{L} \alpha \hat{a}_{l,i,r}}{\sum_{r=1}^{k} a'_{i,r} + L\alpha} \cdot \frac{b'_{r,j} + \sum_{l=1}^{L} \beta \hat{b}_{l,r,j}}{\sum_{j=1}^{v} b'_{r,j} + L\beta} \\ &\quad \cdot \frac{c'_{r,t} + \sum_{l=1}^{L} \gamma \hat{c}_{l,r,t}}{\sum_{t=1}^{n} c'_{r,t} + L\gamma}. \end{aligned} \tag{8}$$

Algorithm 2 TriComp-deComp ($\mathcal{X}^C$)

Input: Current tensor $\mathcal{X}^C \in \mathbb{N}^{u\times v\times \tau}$
Output: Model parameter set $\theta = \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$

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 // Eq. (8)
    end for
  end for
end for
Compute $\mathbf{A}, \mathbf{B}, \mathbf{C}$; //Eq. (9)
$\theta \leftarrow \mathbf{A}, \mathbf{B}, \mathbf{C}$;
$\mathbf{Q}$.deque; // Remove oldest previous parameter
$\mathbf{Q}$.enque(θ); // Insert θ as new previous parameter
return θ;

ここで，$a_{i,r}$，$b_{r,j}$，$c_{r,t}$ は r 番目のグループに entity1 の i 番目，entity2 の j 番目，時刻 t が割り振られた回数を示す．$a'_{i,r}$ 等のプライム符号は，entity1 の i 番目，entity2 の j 番目，時刻 t に割り振られた値が除かれていることを示す．

推定される潜在行列 $\tilde{\mathbf{A}}$，$\tilde{\mathbf{B}}$，$\tilde{\mathbf{C}}$ の各要素は次の式で計算される：

$$\tilde{a}_{i,r} \propto \frac{a_{i,r} + \sum_{l=1}^{L} \alpha \hat{a}_{l,i,r}}{\sum_{r=1}^{k} a_{i,r} + L\alpha}, \quad \tilde{b}_{r,j} \propto \frac{b_{r,j} + \sum_{l=1}^{L} \beta \hat{b}_{l,r,j}}{\sum_{j=1}^{v} b_{r,j} + L\beta}, \quad \tilde{c}_{r,t} \propto \frac{c_{r,t} + \sum_{l=1}^{L} \gamma \hat{c}_{l,r,t}}{\sum_{t=1}^{n} c_{r,t} + L\gamma}. \tag{9}$$

Algorithm 2 は TriComp-deComp の詳細を示している．はじめに，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 $x_{i,j,t}$ に対する潜在グループを決定する．各要素にとっての潜在グループが決定したのち，式 (9) を用いて潜在行列を推定する．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として扱う．モデルの推定後，キュー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．

4.3 TriComp-Compress

候補レジーム θ_c を推定後，TriComp-Compress はレジーム遷移を監視し続け，評価指標である式 (7) に基づいて適切なモデルを選択する．

Algorithm 3 は TriComp-Compress の詳細を示している．レジーム遷移を監視するために，直前レジーム θ_p と候補レジーム θ_c の 2 つを保持する．そして，式 (7) 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．

Algorithm 3 TriComp-Compress ($\mathcal{X}^C, \mathbf{D}, \theta$)

Input: 1. New observation tensor $\mathcal{X}^C \in \mathbb{N}^{u\times v\times \tau}$
2. Previous candidate solution $\mathbf{D} = \{\Theta, S\}$
3. Candidate model parameter set $\theta = \{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$
Output: Updated candidate solution $\mathbf{D}'$

$\theta_c \leftarrow \theta$;
/* Compute $<\mathcal{X}^C; \theta_p>$ and $<\mathcal{X}^C; \theta_c>$; // Eq. (7)
if $<\mathcal{X}^C; \theta_p>$ is less than $<\mathcal{X}^C; \theta_c>$ **then**
  /* Stay on the previous regime θ_p */
  $\theta'_p \leftarrow$ RegimeUpdate (θ_p, θ_c); // Eq. (10)
else
  $\theta_e = \arg\min_{\theta\in\Theta} <\mathcal{X}^C; \theta>$ // Eq. (7)
  if $<\mathcal{X}^C; \theta_c>$ is less than $<\mathcal{X}^C; \theta_e>$ **then**
    /* Shift to the candidate regime θ_c */
    $\Theta' \leftarrow \Theta \cup \theta_c$; $q \leftarrow q + 1$;
    $s_{q+1} = (t, q+1)$;
    $S' \leftarrow S \cup s_{q+1}$;
  else
    /* Shift to the existing regime θ_e*/
    $\theta'_e \leftarrow$ RegimeUpdate (θ_e, θ_c) // Eq. (10)
    $s_e \leftarrow (t, e)$
  end if
end if
return $\mathbf{D}' = \{\Theta' S'\}$;

- 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，既存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．

リアルタイム更新． 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基づいて以下の式で更新される．

$$\tilde{a_{i,r}} \leftarrow \frac{a_{i,r} + \sum_l \alpha \hat{a}_{l,i,r} + \lambda(a^c_{i,r} + \sum_l \alpha \hat{a}^c_{l,i,r})}{\sum_r a_{i,r} + L\alpha + \lambda(\sum_r a^c_{i,r} + L\alpha)},$$
$$\tilde{b_{j,r}} \leftarrow \frac{b_{j,r} + \sum_l \beta \hat{b}_{l,r,j} + \lambda(b^c_{j,r} + \sum_l \beta \hat{b}^c_{l,r,j})}{\sum_r b_{i,r} + L\beta + \lambda(\sum_r b^c_{i,r} + L\beta)}, \tag{10}$$
$$\tilde{c^c_{t,r}} \leftarrow \frac{c_{t,r} + \sum_l \gamma \hat{c}_{l,r,t} + \lambda(c^c_{t,r} + \sum_l \gamma \hat{c}^c_{l,r,t})}{\sum_r c_{t,r} + L\gamma + \lambda(\sum_r c^c_{t,r} + L\gamma)}.$$

ここで，$a^c_{i,r}$ 等の符号 c は，候補レジームの要素を示している．また，本アルゴリズムにおいて学習率 $\lambda > 0$ は固定値とする[*5]．この更新式では，新たに追加される候補レジームの要素は，既存レジームの要素に対してより小さい影響力を持つ．したがって，レジーム間の独立性を保持しながらオンラインに更新することが可能である．これは，モデルの切替えをともなう提案手法に適した更新となっている．

定理 1　各カレントテンソルにおいて TriComp は単位

[*5] 本論文では $\lambda = 0.1$ とする．

時間あたり最小 $O(N)$，最大 $O(q+N)$ 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（$N=\sum_{i,j,t} x_{i,j,t}$）．

証明 1　各時刻において，TriComp はまず TriComp-deComp を行う．$\mathcal{X}^C$ 内における各イベントエントリにおいて，潜在グループ z を決定する．グループ数を k，学習の反復回数を $\#iter$ とすると，この手順は $O(\#iter \times kN)$ 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，TriComp-deComp の計算時間は $O(N)$ である．次に，TriComp-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，$O(q)$ の計算時間を要する．全体として，TriComp はこれらの 2 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 $O(q+N)$ の計算時間を要する．

5. 評価実験

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

- 提案手法から得られる要約の有効性
- 提案手法の要約精度
- 複合イベントテンソル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 3.5 GHz quad core CPU，192 GB 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 2 つのデータセットについて，**表 2** に示す[*6]．

- NY-Taxi[*7]：2020 年 1 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 (乗車エリア ID, 降車エリア ID, 1 時間刻みの乗車時間) の 3 つの属性から構成されている．
- NY-Bike[*8]：2015 年のニューヨーク市自転車シェアリングサービスの利用履歴．各記録は，(利用ユーザ世代, 利用開始エリア ID, 1 時間刻みで取得した利用開始時間) の 3 つの属性値を持つ．利用ユーザ世代は，10 歳から 100 歳までを 5 歳ごとに分類（離散化）し，それぞれが該当する年代を属性情報とした．

表 2　データセットの概要
Table 2　Dataset description.

ID	Dataset	entity1	entity2	time	sparsity (%)
#1	NY-Taxi	262	263	4,368	98.262
#2	NY-Bike	19	488	8,760	93.263

[*6] sparsity は $(1-\frac{\#observation}{u\times v\times n})\times 100$ によって求められる．
[*7] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[*8] https://www.citibikenyc.com/system-data

5.1 Q1. 提案手法の有効性

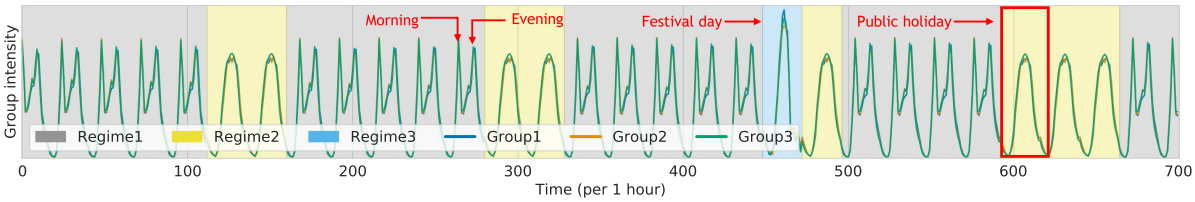
NY-Taxi. NY-Taxi データセットの結果は 1 章で示したとおりである（図 2）．この結果では，直感的な社会活動と一致するような，9 つのセグメントと 2 つのレジームの検出に成功している（$q=2$）．ここで，平日の特徴をとらえたと考えられるレジーム 1（平日レジーム）を取り上げる．図 2 (a) の平日レジームでは，グループ 1（青色）が 1 日の始めと終わりに高いグループ関連度を持つことに対し，グループ 2（橙色），3（緑色）は 1 日の終わりにかけてグループ関連度が高くなることを示している．これは，グループ 1 が 1 日の始めと終わりに多く利用されているグループであり，グループ 2，3 は夜にかけて利用が増加するようなグループであることを示している．潜在グループはすべての属性間で共通しているため，上記の傾向を持つユーザは図 2 (b)，(c) の各色で示したエリアで乗車・降車していることが分かる．重要な点として，提案手法は事前知識を必要とせずに自動で上述の特徴を抽出し，リアルタイムに処理を行う．

NY-Bike. **図 4** は，NY-Bike での解析結果を示す．まず，直感に従うような類似時系列パターンの検出に関する有効性について述べる．図 4 (a) において，提案手法は 3 種類のレジームを自動的に検出している．最終的に得られたこれらのレジームの変化点と割当てから，それぞれのレジームが平日，休日，祭日と一致していることが分かる．ここで重要な点として，提案手法は，時刻 480 の祭日や時刻 600 の休日といった，不規則なパターンを正確にとらえている．TriComp におけるレジームの検出は出現頻度に依存しないため，祭日のような突然発生したイベントもとらえることが可能である．

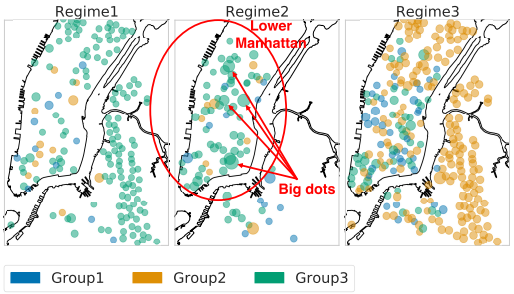
続いて，検出されたそれぞれのレジームに関して述べる．

レジーム 1（平日レジーム）：早朝と夕方に強いグループ関連度を示しており，シェアリングサービスが通勤のためによく利用されているのではないかと推測できる（図 4 (a)）．図 4 (b) では，小さな点が地図全体に点在している．これは，特定のエリアで集中して利用されているわけでなく，様々なエリアでサービスが利用されていることを示している．

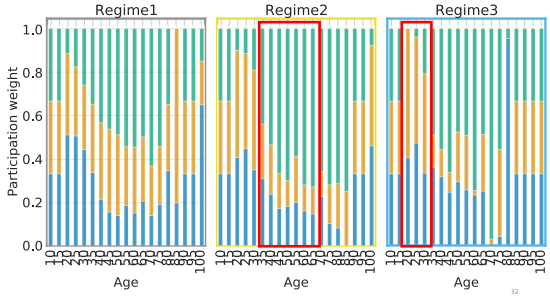
レジーム 2（休日レジーム）：図 4 (a) において昼ごろに集中を示している．また，図 4 (b) において，いくつかの大きな点が見られ，点群は赤円の範囲内に集中している．これは，休日において，他のエリアと比較して多くのユーザが，特定のエリア（Lower Manhattan）でシェアリングサービスを利用しているということを示している．図 4 (c) において，平日レジームと休日レジームを比較すると，大きな点を示していたグループ 3（緑色）と中年層が強い関連度を



(a) Regime assignments and time-wise group intensity



(b) Start station-specific groups



(c) User's age-specific participation weights

図 4　(a) レジーム割当て結果と時間ごとのグループ関連度 (b) 3 つの潜在グループとそれらに対する各エリアの関連度（点の大きさ）．0.4 以下の関連度は可視化していない．(c) 潜在グループに対する各世代の参加強度

Fig. 4　Modeling power of TriComp for bicycle ride-share events: (a) Regime assignment and time-wise group intensity (Regime 1: weekday, Regime 2: weekend, Regime 3: festival day). (b) Three latent groups (three colors) and their station-wise participation weights (size of dot). Too low degrees are not shown (< 0.4). (c) Participation weights of each generation in their respective groups.

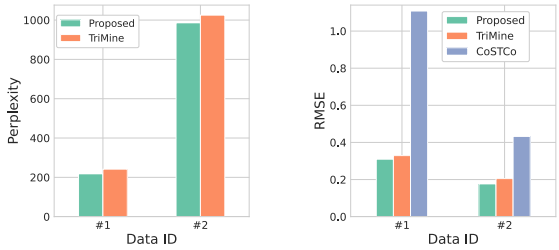
持っていることが分かる．これは休日の Lower Manhattan において，中年層が主要なユーザであることを示している．

レジーム 3（祭日レジーム）：この日は，ニューヨーク市で式典やパレードが催されていた日と一致している．こうしたイベントは短時間の混雑を引き起こすため，昼ごろに休日より強い集中を示している．図 4 (b) において，グループ 2（橙色）が多く出現し，図 4 (c) では，若年層とグループ 2 の関連度が大きくなっている．これは，若年層ユーザがグループ 2 のエリアで多く利用していたことを示している．祭日において，休日では主要なユーザではなかった若年層ユーザが，このサービスに強い関心を持っていることが分かる．

まとめると，提案手法は，既存手法では達成し得ない，効果的かつ有益な洞察を提供する．

5.2　Q2. 提案手法の精度

続いて，提案モデルによる要約の精度について検証する．提案手法の目的は，大規模かつ複雑なイベントテンソルストリームを効果的に表現可能な要約情報の抽出である．そこで，得られた要約情報を用いてデータを再構成した際の RMSE と perplexity を評価した．RMSE は，モデルから得られた予測値と実際の値との誤差を示す．perplexity は，確率モデルや確率分布の性能を評価する尺度であり，モデルを用いて対象データを予測した際の精度を示す．両者とも，低い値は高いモデル精度を意味する．比較手法として



(a) Average perplexity　　(b) RMSE

図 5　再構成結果の Average perplexity と RMSE

Fig. 5　Accuracy of TriComp (Average perplexity, RMSE).

以下のテンソル分解アルゴリズムを用いる．

TriMine [17] 3 つ組で構成されるイベント群においてオフラインに分解を行う既存手法である．

CosTCo [22] 大規模スパースデータのための最新手法で，ニューラルネットワークに基づきテンソル分解，テンソル補完を行う．学習は文献 [22] に準拠し，最適化には Adam アルゴリズムを使用した．

図 5 では，毎時刻，長さ 24 の $\mathcal{X}^C$ を与えた場合の，再構成テンソルとオリジナルテンソルとの，平均 perplexity と平均 2 乗誤差（RMSE）を示している．両指標において，低い値は高いモデル精度を意味する．提案手法は，リアルタイムに類似時系列パターン（レジーム）を適切にとらえ，効果的にモデルを更新することが可能であるため，全体として，既存のオフライン手法と比較し高い精度を示してい

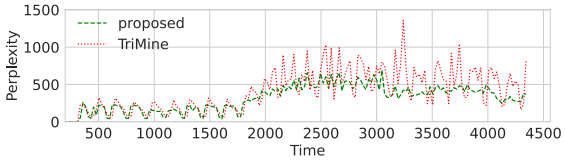
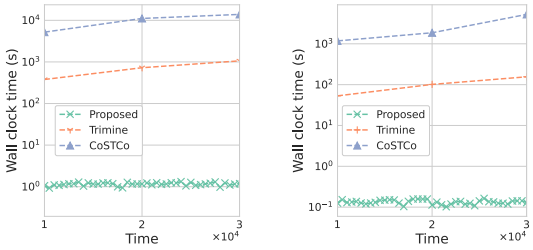


図 6　各時刻における提案手法の精度
Fig. 6　Accuracy of TriComp per time point.



(a) #1：NY-Taxi　　(b) #2：NY-Bike

図 7　各時刻における TriComp の計算コスト
Fig. 7　Wall clock time of TriComp.

る．CosTCo はスパースで大規模なテンソルを処理することが可能であるが，潜在的な時系列変化がともなう，複合イベントテンソルストリームに最適化されていない．

また，NY-Taxi データセットにおける各時刻の perplexity を図 6 に示す．NY-Taxi データセットでは，全体長の中間時点から傾向が変わる．既存手法の 1 つである TriMine では，このような急な傾向の変化を適切にとらえることができない．一方，提案手法は，効果的な要約とモデルの切替えによって，傾向が変化した後も低い値を示し続けている．

5.3　Q3. 提案手法の計算コスト

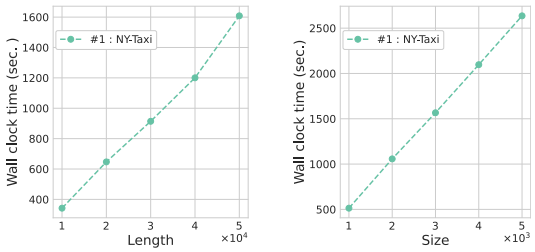
図 7 は各時刻における計算コストを，既存手法と比較したものである．各データセットにおいて，提案手法は逐次的な更新により，既存手法に比べて最大 4 桁の高速化を実現している．

図 8 は入力テンソルのサイズ（データ長，entitiy1 の総数）を変化させたときの計算コストを示している．テンソル分解に基づく高速かつ効率的なモデル推定によって，すべての実験においてテンソルサイズに線形な計算量であり，大規模なイベントテンソルストリームの解析に適した手法である．

6.　むすび

本論文では，大規模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 TriComp を提案した．

(1) 複雑なイベントデータから，時系列パターンや属性内に存在するグループといった潜在的な特徴を発見する．
(2) 特徴抽出は自動的に行われ，データの解釈を助けるような要約情報をリアルタイムに提供する．



(a) Vs. duration　　(b) Vs. entity1

図 8　テンソルサイズに対する平均計算時間
Fig. 8　Average wall clock time vs. tensor stream size, i.e., duration and the number of entity1.

(3) 半無限長となる複合イベントテンソルストリームを一定の計算時間で処理する．

実データを用いた評価実験では，複雑なイベントテンソルストリームから自動で要約情報を抽出することを確認した．また，計算コストはデータストリームの長さに依存せず，従来の手法と比較して大幅に性能が向上していることを示した．

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．

参考文献

[1] Nehme, R.V., Rundensteiner, E.A. and Bertino, E.: Tagging stream data for rich real-time services, *Proc. 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, *Proc. 18th International Conference on World Wide Web*, pp.21–30 (2009).
[3] Ho, J.C., Ghosh, J. and Sun, J.: Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization, *Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.115–124 (2014).
[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, *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1265–1274 (2015).
[5] Harshman, R.A. et al.: Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis (1970).
[6] Tucker, L.R.: Some mathematical notes on three-mode factor analysis, *Psychometrika*, Vol.31, No.3, pp.279–311 (1966).
[7] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: The Web as a Jungle: Non-Linear Dynamical Systems for Co-evolving Online Activities, *WWW* (2015).
[8] Takahashi, T., Hooi, B. and Faloutsos, C.: AutoCyclone: Automatic Mining of Cyclic Online Activities with Ro-

bust Tensor Factorization, *WWW*, pp.213–221 (2017).
[9] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: Non-Linear Mining of Competing Local Activities, *WWW* (2016).
[10] Song, H.A., Hooi, B., Jereminov, M., Pandey, A., Pileggi, L. and Faloutsos, C.: PowerCast: Mining and forecasting power grid sequences, *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp.606–621, Springer (2017).
[11] Wen, Q., Gao, J., Song, X., Sun, L., Xu, H. and Zhu, S.: RobustSTL: A robust seasonal-trend decomposition algorithm for long time series, *Proc. AAAI Conference on Artificial Intelligence*, Vol.33, pp.5409–5416 (2019).
[12] Wen, Q., Zhang, Z., Li, Y. and Sun, L.: Fast RobustSTL: Efficient and Robust Seasonal-Trend Decomposition for Time Series with Complex Patterns, *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2203–2213 (2020).
[13] Hooi, B., Shin, K., Liu, S. and Faloutsos, C.: SMF: Drift-aware matrix factorization with seasonal patterns, *Proc. 2019 SIAM International Conference on Data Mining*, pp.621–629, SIAM (2019).
[14] Xiong, L., Chen, X., Huang, T.-K., Schneider, J. and Carbonell, J.G.: Temporal collaborative filtering with Bayesian probabilistic tensor factorization, *Proc. 2010 SIAM International Conference on Data Mining*, pp.211–222, SIAM (2010).
[15] Schein, A., Paisley, J., Blei, D.M. and Wallach, H.: Bayesian Poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts, *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1045–1054 (2015).
[16] Schein, A., Zhou, M., Blei, D. and Wallach, H.: Bayesian Poisson Tucker decomposition for learning the structure of international relations, *International Conference on Machine Learning*, pp.2810–2819, PMLR (2016).
[17] 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).
[18] Dalleiger, S. and Vreeken, J.: Explainable Data Decompositions, *AAAI*, pp.3709–3716 (2020).
[19] Liu, B., He, L., Li, Y., Zhe, S. and Xu, Z.: Neuralcp: Bayesian multiway data analysis with neural tensor decomposition, *Cognitive Computation*, Vol.10, No.6, pp.1051–1061 (2018).
[20] Kim, D., Park, C., Oh, J., Lee, S. and Yu, H.: Convolutional matrix factorization for document context-aware recommendation, *Proc. 10th ACM Conference on Recommender Systems*, pp.233–240 (2016).
[21] Socher, R., Chen, D., Manning, C.D. and Ng, A.: Reasoning with neural tensor networks for knowledge base completion, *Advances in Neural Information Processing Systems*, Vol.26, pp.926–934 (2013).
[22] Liu, H., Li, Y., Tsang, M. and Liu, Y.: CoSTCo: A Neural Tensor Completion Model for Sparse Tensors, *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.324–334 (2019).
[23] Dabrowski, J.J., Rahman, A., George, A., Arnold, S. and McCulloch, J.: State space models for forecasting water quality variables: An application in aquaculture prawn farming, *Proc. 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.177–185 (2018).
[24] De Livera, A.M., Hyndman, R.J. and Snyder, R.D.: Forecasting time series with complex seasonal patterns using exponential smoothing, *Journal of the American Statistical Association*, Vol.106, No.496, pp.1513–1527 (2011).
[25] Li, L., Prakash, B.A. and Faloutsos, C.: Parsimonious linear fingerprinting for time series, Technical Report, *Proc. VLDB Endowment*, Vol.3, No.1, Carnegie-Mellon University Pittsburgh PA School of Computer Science (2010).
[26] Matsubara, Y., Sakurai, Y. and Faloutsos, C.: AutoPlait: Automatic Mining of Co-evolving Time Sequences, *SIGMOD* (2014).
[27] Hallac, D., Vare, S., Boyd, S. and Leskovec, J.: Toeplitz Inverse Covariance-Based Clustering of Multivariate Time Series Data, *KDD*, pp.215–223 (2017).
[28] Kawabata, K., Matsubara, Y. and Sakurai, Y.: Automatic sequential pattern mining in data streams, *Proc. 28th ACM International Conference on Information and Knowledge Management*, pp.1733–1742 (2019).
[29] Honda, T., Matsubara, Y., Neyama, R., Abe, M. and Sakurai, Y.: Multi-aspect Mining of Complex Sensor Sequences, *ICDM*, pp.299–308 (2019).
[30] Kawabata, K., Matsubara, Y., Honda, T. and Sakurai, Y.: Non-Linear Mining of Social Activities in Tensor Streams, *Proc. 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp.2093–2102 (2020).
[31] Porteous, I., Newman, D., Ihler, A., Asuncion, A., Smyth, P. and Welling, M.: Fast collapsed Gibbs sampling for latent Dirichlet allocation, *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.569–577 (2008).

中村 航大

2020 年熊本大学工学部情報電気電子工学科卒業．現在，大阪大学産業科学研究所および大学院情報科学研究科博士前期課程に在籍．大規模時系列データマイニングの研究に従事．日本データベース学会学生会員．

松原 靖子

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

川畑 光希

2016 年熊本大学工学部情報電気電子工学科卒業．2018 年同大学院博士前期課程修了．2021 年大阪大学大学院情報科学研究科情報システム工学専攻博士後期課程修了．博士（情報科学）．2019 年大阪大学情報科学研究科日本学術振興会特別研究員（DC2）．2021 年 4 月より大阪大学産業科学研究所助教．DEIM Forum 2016 最優秀論文賞，WebDB Forum 2018 最優秀論文賞，学生奨励賞，企業賞，2019 年度コンピュータサイエンス領域奨励賞（データベースシステム），等受賞．データマイニング，データストリーム処理の研究に従事．日本データベース学会会員．

梅田 裕平

2005 年九州大学理学部数学科卒業．2007 年同大学院数理学府理学専攻修士課程修了．2009 年同大学院数理学府理学専攻博士課程修了．博士（機能数理学）．2009 年から九州大学特別研究員．2010 年株式会社富士通研究所入社．2021 年より富士通株式会社人工知能研究所プロジェクトマネージャー．2015 年度計測自動制御学会論文賞等受賞．人工知能，機械学習，データ処理，時系列解析技術の研究に従事．応用数理学会，人工知能学会，IEEE 各会員．

和田 裕一郎

2009 年東京工業大学情報科学科卒業．2011 年同大学院博士前期課程修了．2019 年名古屋大学大学院情報科学研究科博士後期課程修了．同年 10 月より，東京工業大学特別研究員．2020 年富士通研究所研究員と理研 AIP 客員研究員．機械学習，特に教師なし学習に関する研究に従事．

櫻井 保志

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．2019 年より大阪大学産業科学研究所産業科学 AI センターセンター長・教授．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，IEEE，電子情報通信学会，日本データベース学会各会員．

（担当編集委員　小林 亜樹）