The University of Osaka
Institutional Knowledge Archive

Title	車両走行センサデータからの自動パターン検出
Author(s)	本田, 崇人; 松原, 靖子; 根山, 亮 他
Citation	情報処理学会論文誌データベース (TOD) . 2016, 9(3), p. 1-13
Version Type	VoR
URL	https://hdl.handle.net/11094/93128
rights	©2016 Information Processing Society of Japan
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

情報処理学会論文誌　データベース　Vol.9　No.3　1–13 (Sep. 2016)

推薦論文

車両走行センサデータからの自動パターン検出

本田 崇人[1,a)]　松原 靖子[1]　根山 亮[2]　櫻井 保志[1]

受付日 2015年12月20日, 採録日 2016年4月8日

概要：本論文では，車両走行データのための自動パターン検出手法である TrailMarker について述べる．TrailMarker は，位置情報をともなう様々な車両走行センサデータが与えられたときに，おのおのの道路や場所における車両走行の特徴を抽出し，それらの情報を統計的に要約，表現する．すなわち，走行データに基づく高度な道路地図情報を提供する．具体的に提案手法は，(a) 車両走行データをテンソルとして表現した後，そこから複数の部分シーケンスに共通する主要な走行パターンを抽出する．(b) その際の計算量は入力データのサイズに対して線形である．さらに，最も重要な点として，(c) 提案手法はパラメータに依存しない．すなわち，事前情報の付与またはパラメータのチューニングを行うことなく，大規模車両走行データの特徴抽出とパターン検出を自動で行うことができる．実データを用いた実験では TrailMarker が様々な車両走行データの中から主要パターンや外れ値シーケンスを効果的かつ効率的に検出することを確認した．

キーワード：車両走行センサデータ，自動パターン検出，地理情報テンソル

Fully Automatic Mining of Large Geographical Complex Sequences

Takato Honda[1,a)]　Yasuko Matsubara[1]　Ryo Neyama[2]　Yasushi Sakurai[1]

Received: December 20, 2015, Accepted: April 8, 2016

Abstract: In this paper we present TrailMarker, a fully automatic mining algorithm for geographical complex sequences. Our method has the following properties: (a) effectiveness: it operates on large collections of time-series, and finds similar segment groups that agree with human intuition; (b) scalability: it is linear with the input size, and thus scales up very well; and (c) TrailMarker is parameter-free, and requires no user intervention, no prior training, and no parameter tuning. Extensive experiments on real datasets demonstrate that TrailMarker does indeed detect meaningful patterns effectively.

Keywords: vehicle sensor data, automatic mining, geographical tensor

1. はじめに

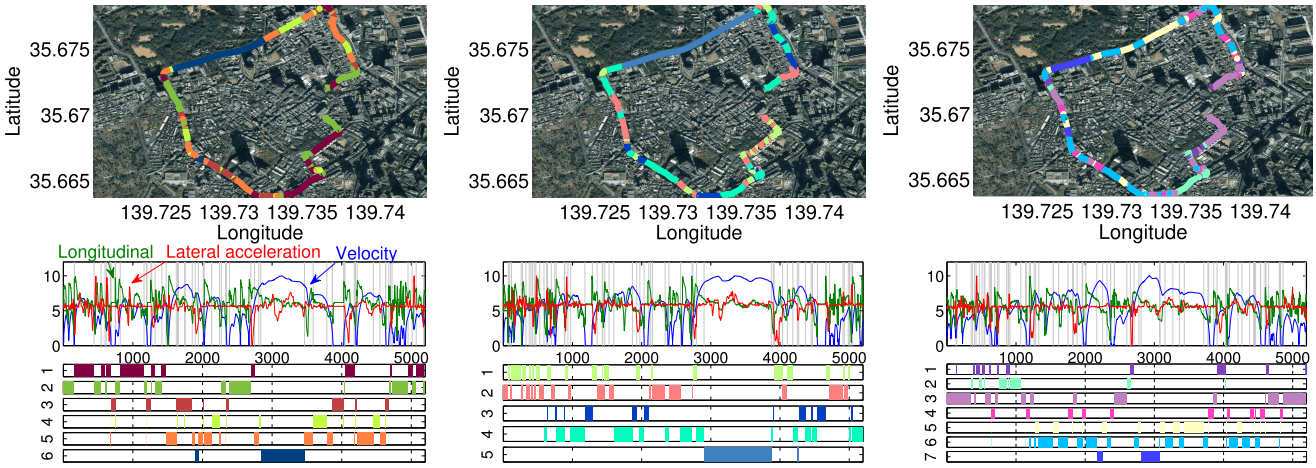
車両走行センサデータの解析は，安全で快適な自動車走行のための技術向上，ならびに情報ネットワークを活用した新たな運転サービスの提供のために非常に重要な課題となっている．本論文では，大規模な車両走行センサデータを対象とし，重要な車両走行パターンの抽出，もしくは異常パターンの検出を自動的に行うことを目的とする．より具体的には，様々な道路，多数の車両，複数のセンサからのデータが与えられたとき，これら大規模な車両走行センサデータを多次元の地理情報テンソルとして扱い，すべての要素を統合的に解析し，データ全体を表現する要約情報を抽出する．そして，走行データに基づく高度な道路地図情報を提供する．

一般に，実際に生成される車両走行センサデータは，複数の異なるトレンドやパターンを持つことが多い．たとえ

1　熊本大学大学院自然科学研究科
Kumamoto University, Kumamoto 860–8555, Japan
2　トヨタ IT 開発センター
TOYOTA Info Technology Center Co., Ltd., Minato, Tokyo 107–0052, Japan
a)　takato@dm.cs.kumamoto-u.ac.jp

本稿の内容は 2015 年 11 月の WebDB フォーラム 2015 で発表され，同シンポジウムプログラム委員会により情報処理学会論文誌データベースへの掲載が推薦された論文である．

© 2016 Information Processing Society of Japan



(a) H-レジーム 1 (トリップ 数: 9/31)　(b) H-レジーム 2 (トリップ 数: 21/31)　(c) 外れ走行 (トリップ 数: 1/31)

図 1　車両走行データにおける TrailMarker の出力例（Y コース，総トリップ数：31）

Fig. 1　TrailMarker "automatically" identifies driving patterns (e.g., turn) and features (e.g., driver) of vehicle sensor data, as well as the positions of the all cut points (Y course, 31 trips).

ば，一般的な道路では，曲がり角や信号，車線変更など様々な走行パターンを持つ．また，同じ道路であっても時間帯や運転者によって走行パターンは異なる．ある道路を走行している際，安定した走行と異常，つまり危険な走行など，様々な走行が見られる．ここで，これらの走行パターンを本論文では「V-レジーム（V-regime)」，走行グループを「H-レジーム（H-regime)」と呼ぶ．本研究では，大規模な地理情報テンソルの中から，これらの異なるトレンドを発見し，すべての車両走行パターンを表現する手法として，TrailMarker を提案する．

本論文で扱う問題は以下のとおりである．

問題：車両走行センサデータ集合 $\mathcal{X}$ が与えられたとき，$\mathcal{X}$ を表現する車両走行パターンを抽出する．より具体的には

(1) $\mathcal{X}$ の中のパターンの変化点を発見し，部分シーケンス集合（セグメント）に分割し，
(2) セグメントの共通パターンを検出するとともに，
(3) 類似した車両走行シーケンスをグループ化する．
(4) さらに重要な点として，これらの処理は高速かつ自動で行う．

具体例．　図 1 は，赤坂 Y コースの車両走行センサデータと TrailMarker の出力結果例である．この車両走行のセンサデータ集合には合計 31 の多次元シーケンスが含まれており，シーケンスの各要素は 3 次元の値から構成され，それぞれの次元が，速度（青），左右加速度（赤），前後加速度（緑）を示している．図 1 (a)，(b)，(c) はおのおの類似した車両走行シーケンスのグループ（H-レジーム）を示しており，各グループにそれぞれ 9，21，1 つのシーケンスが割り当てられている．図 1 (a) の上段は TrailMarker の出力結果から，1 つの典型的なシーケンスを地図上にプロットしたものであり，下段はグループにおける 1 つの典型的なシーケンスと，TrailMarker が自動抽出した 6 つのセグメント共通パターン（V-レジーム）をグループの代表として示している．同一の V-レジームに含まれるセグメントは同一の色で表現されている．

提案手法は，ハンドル操作，加速や減速，停止など，車両走行の様々な共通パターンである V-レジームを抽出すると同時に，慎重な走行（図 1 (a)），スムーズで慣れた走行（図 1 (b)），経験の浅い走行（図 1 (c)）などの H-レジームの発見，すなわち車両走行のグループ化も行うことができる．

ここで最も重要なこととして，TrailMarker はこれらの走行パターンに関する事前知識を必要とせず，適切な数の V-レジーム，H-レジームを自動的に把握することができる．

1.1　自動抽出手法の重要性

クラスタリング [9]，セグメンテーション [5], [22]，類似シーケンス探索 [16], [18] などセンサデータを対象とした研究課題は数多く存在するが，これらの先行研究は基本的にすべてパラメータの設定やチューニングを必要とする．セグメントの個数やエラーの閾値など，ユーザに様々なパラメータ入力の負担を強いるだけでなく，出力結果にも大きな影響を与える．特にビッグデータの解析において，ユーザの手を介したパラメータ設定は多くの時間的コストを必要とするため，自動処理技術は必要不可欠な要素である．

1.2　本論文の貢献

本論文では車両走行センサデータ集合を多次元の地理情報テンソルに変換し，縦方向（Vertical）や横方向（Horizontal）に分割しながら，複数の観点からすべての要素を

© 2016 Information Processing Society of Japan

統合的に解析する．提案手法 TrailMarker は以下の特長がある．

(1) すべての車両走行シーケンスにおいて共通する部分シーケンスパターンの個数を求め，おのおののパターンの特徴をモデル（V-レジーム）として表現する．
(2) V-レジームのモデルを用いて類似した車両走行シーケンスのグループ化を行う．提案するコスト関数に基づいて適切なグループ数を求めながら，各グループの特徴（H-レジーム）をとらえる．
(3) TrailMarker はパラメータ設定を必要としない．ユーザの介入を必要とせず，適切な V-レジームの数，H-レジームの数，変化点の数を，自動的に発見することができる．
(4) 縦方向と横方向の分割と特徴抽出を交互に行いながら，効率的にテンソルの解析を行う．計算コストは入力データの長さ，車両走行データの数に対して線形である．

2. 関連研究

関連研究は以下の 3 つに分類される．

パターン発見． センサデータの解析に関する研究は，時系列マイニングなど様々な分野で進められている [2], [10], [11], [13]．自己回帰モデル（AR: autoregressive model），線形動的システム（LDS: linear dynamical systems）は代表的な技術であり，これらに基づくセンサデータの解析と予測手法が数多く提案されている [19]．また，本論文と関連するテンソル解析についても，Web 情報を解析するための様々な手法が提案されている [12], [14]．Li らは文献 [8] において，欠損を含む大規模時系列シーケンス集合のためのアルゴリズムである DynaMMo を提案している．DynaMMo は LDS に基づき，時系列データのパターンを発見し，シーケンスのセグメント化の能力を持つ．Rakthanmanon らは文献 [16] において，兆単位（“trillions”）の時系列シーケンスを対象とした DTW の類似探索問題を扱っている．著者らの先行研究 [20] では，時系列データのパターン発見とセグメント化，クラスタリングを完全自動で行う手法を提案しているが，テンソルデータを対象としておらず，扱う問題が異なる．

確率モデル． 隠れマルコフモデル（HMM: Hidden Markov model）は音声認識を含む様々な分野において，時系列データ処理手法として広く利用されている [23]．HMM に基づく大規模時系列シーケンスのための研究として，文献 [7] では，RFID センサから生成された時系列のマルコフストリームを対象としたイベント問い合わせの手法が提案され，一方，文献 [4] では大規模 HMM データ集合のための高速探索アルゴリズムが扱われている．最新の研究として，Wang ら [22] は文献 [5] を改良し，pHMM（pattern-based hidden Markov model）を提案している．pHMM は時系列データのセグメント化とクラスタリングのための動的モデルであり，シーケンスをマルコフモデルに基づいて線形のセグメントに分割する能力を持つ．これらの手法は，時系列データの複雑な動的パターンを表現する能力があるが，その一方で，高度なパラメータチューニングや，モデルの構造の定義などが必要となり，さらに，これらの手法は大規模センサデータの解析を想定していない．

また，テンソル解析手法として，Matsubara ら [12], [14] は TriMine（Fast mining and forecasting of complex time-stamped events）を提案している．TriMine は大規模イベントデータから潜在的なトレンドやパターンを検出可能であるが，web クリックデータを対象としており，本手法とは扱う問題が異なる．

情報抽出とクラスタリング． 情報抽出とクラスタリングの手法は CLARANS [15]，BIRCH [24]，TRACLUS [6] を含め，様々なものが提案されている．パラメータフリーな情報解析手法としては，OCI [1] がある．OCI は，外れ値を含む点集合のクラスタリングのための手法である．さらに，文献 [3], [21] においては，MDL の概念を用いて情報要約とクラスタリング問題を扱っている．

3. コンセプトと問題定義

ここでは本論文で必要な概念について定義を行う．本研究において扱う車両走行データは時間，場所（緯度，経度），センサによる計測値から構成され，トリップごとに毎時刻収集される．トリップとは，特定の車両による 1 つの目的を持った出発地から到着地までの移動を指す．本論文では，場所ごとの車両走行の特徴を抽出するため，すべての道路には**ゾーン**と呼ぶ小さな区域を設ける．そして，各ゾーンは 1 カ所の計測場所を有する[*1]．したがって車両走行データは $(trip, zone, object)$ のように構成される要素の一連のシーケンスとして表現される複合データである．ここで，トリップ（$trip$）とゾーン（$zone$）の総数をそれぞれ w と n とする．そして $object$ は各種センサによる計測値を表しており，d 次元ベクトルとして表現される[*2]．本論文ではこのようなデータを地理情報テンソルと呼ぶ．

定義 1（地理情報テンソル） $\mathcal{X} \in \mathbb{R}^{w \times d \times n}$ を地理情報テンソルとする．$\mathcal{X}$ の要素 $x_{i,z,j}$ は，i 番目のトリップにおけるゾーン z の j 番目のセンサノードの計測値を示している．

地理情報テンソル $\mathcal{X}$ から i 番目のトリップの情報，すなわちセンサノードの計測値を取り出したとき，トリップ i の地理複合シーケンスと呼ぶ．

*1 1 つの ゾーンが複数の計測場所を持つ場合には，ゾーンの中心点に近い計測値を選択するか，中心からの距離に基づく重み付き平均をとることにより求めることができる．
*2 本論文ではセンサによる計測値として，速度，前後加速度，左右加速度を用い，またゾーンとして道路を 1 m 間隔に区分している．

© 2016 Information Processing Society of Japan

定義 2（地理複合シーケンス） $\boldsymbol{X}_i = \{\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,n}\}$ をトリップ i の長さ n の地理複合シーケンスとする．$\boldsymbol{x}_{i,z} = \{x_{i,z,j}\}_{j=1}^{d}$ はゾーン z における計測値である．
すなわち，$\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_w\}$ である．図 1 は車両走行データ，すなわち地理複合シーケンスの例であり，各ゾーンにおける d 次元のオブジェクトシーケンスを示している．

1 つの地理複合シーケンス $\boldsymbol{X}$ が与えられたとき，$\boldsymbol{X}$ を m 個のセグメント $s_1, \ldots, s_m$ に分割してその特徴をとらえる（5.2 節を参照）．s_i は i 番目のセグメントの範囲を表し，ある 1 つの走行パターンが現れる範囲を表している．これはセグメントの開始点 t_s，終了点 t_e，トリップ番号 j で構成され（つまり，$s_i = \{t_s, t_e, j\}$），各セグメントは重複がないものとする．そして，発見したこれらのセグメント集合 $\boldsymbol{s} = \{s_1, \ldots, s_m\}$ を類似セグメントのグループに分類する．すなわち，類似した走行パターン（車線変更，信号停止，右折など）を表すセグメント同士のグループ化を行う．

定義 3（V-レジーム） r を最適なセグメントグループの個数とする．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．これらグループを V-レジーム（V-regime）と呼び，それぞれの V-レジームは統計モデル $\boldsymbol{\theta}_i$ $(i = 1, \ldots, r)$ として表現される．
V-レジームは後述（5.2 節）のアルゴリズム V-Split によって作成されるセグメントグループであり，たとえば，図 1 (a) において，シーケンスは $r = 6$ 個の V-レジームから構成され，それぞれのセグメントが $r = 6$ 個の V-レジームのうちの 1 つに割り当てられる．

定義 4（セグメントメンバーシップ） 地理複合シーケンス $\boldsymbol{X}$ が与えられたとき，$\boldsymbol{v} = \{v_1, \ldots, v_m\}$ を，m 個の整数列とし，v_i を i 番目のセグメントが所属する V-レジームの番号とする $(1 \leq v_i \leq r)$．
図 1 (a) では，1 番目のセグメントは 2 番目の V-レジームに，2 番目のセグメントは 1 番目の V-レジームにそれぞれ所属する．つまり，この場合のセグメントメンバーシップは $\boldsymbol{v} = \{2, 1, 2, 1, ...\}$ となる．

次に，複数トリップからの特徴抽出について考える．$\mathcal{X} = \{\boldsymbol{X}_i, \ldots, \boldsymbol{X}_w\}$ を w 個のトリップの地理情報テンソルとする．本研究の目的は大規模な $\mathcal{X}$ が与えられたときに，(a) おのおののトリップのグループ化と，(b) 各グループにおけるトリップシーケンスのセグメンテーション，それら両方を行いながら複数のトリップシーケンスに共通する特徴を高速かつ自動で抽出することである．そこで，本研究ではセグメンテーションだけでなく，$\mathcal{X}$ を g 個のトリップグループに分割してパターン抽出を行う．

定義 5（H-レジーム） g を最適なトリップグループの個数とする．それぞれのトリップはトリップグループの 1 つに割り当てられる．これらグループを H-レジーム（H-regime）と呼び，それぞれの H-レジームはコア $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_g\}$ として表現される．
H-レジームは後述（5.3 節）のアルゴリズム H-Split によって作成されるトリップグループである．たとえば，図 1 において，地理情報テンソルは $g = 3$ 個の H-レジームから構成され，それぞれのトリップが $g = 3$ 個の H-レジームの内の 1 つに割り当てられる．$\boldsymbol{\phi}_i$ は i 番目の H-レジームのコアであり，i 番目のグループを代表するトリップが，各ゾーンにおいてどのモデル $\boldsymbol{\theta}_j$ $(j = 1, \ldots, r)$ を用いて表現されているのかを示している．すなわち，$\boldsymbol{\phi}_i$ は長さ n の整数列であり，各ゾーンが所属する V-レジームの番号を表す．そして，セグメントグループを表現する V-レジームは，1 つの H-レジーム内でのみ共有される．

定義 6（トリップメンバーシップ） 地理情報テンソル $\mathcal{X}$ が与えられたとき，$\mathcal{H} = \{h_1, \ldots, h_w\}$ を，w 個の整数列とし，h_i を i 番目のトリップが所属する H-レジームの番号とする $(1 \leq h_i \leq g)$．

本論文で取り組む問題を以下のように定義する．

問題 1 地理情報テンソル $\mathcal{X}$ が与えられたとき，すべてのトリップの地理複合シーケンス $\boldsymbol{X}_i$ $(i = 1, \ldots, w)$ を表現するような以下の情報を抽出する．

(1) 各セグメントの位置とセグメント総数：
$\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_w, \boldsymbol{m}\}$

(2) V-レジームの総数 r とセグメントメンバーシップ：
$\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_w\}$

(3) H-レジームの総数 g とトリップメンバーシップ：
$\mathcal{H} = \{h_1, \ldots, h_w\}$

(4) r 個の V-レジームを表現するモデルパラメータ集合：
$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}_{r \times r}\}$

(5) g 個の H-レジームのコア集合：
$\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_g\}$

ここで，$\boldsymbol{\Delta}_{r \times r}$ は V-レジーム遷移行列，$\boldsymbol{m} = \{m_1, \ldots, m_w\}$ は各トリップにおけるセグメント数である．上記のすべての情報は最小記述長原理に基づくコスト関数（式 (2)）を最小化するものを選ぶ．
本論文では，V-レジームを表現するモデルパラメータ集合 $\boldsymbol{\Theta}$ を，r 個の隠れマルコフモデル（HMM: hidden Markov model），$\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r\}$，として表現する[*3]．

問題 1 で示したとおり，本論文の目的は，$\mathcal{X}$ の特徴を抽出し，すべてのパターンを表現するパラメータ集合を発見することである．

定義 7 $\mathcal{X}$ を表現する全パラメータ集合 $\mathcal{C} = \{r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}\}$ を候補解と呼ぶ．候補解 $\mathcal{C}$ は，セグメント集合，各セグメント，各トリップの V-レジーム，H-レジームへの割当て，V-レジームを表現する確率モデル，H-レジームのコア，これらすべてを表現する．

表 1 に主な記号と定義を示す．結論として，本論文の目的は最適な解 $\mathcal{C}$ を発見することである．ここで非常に重要

*3 提案する枠組みは，HMM 以外のモデルに適用することも可能である．

© 2016 Information Processing Society of Japan

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

記号	定義
テンソル	
n	地理複合シーケンスの長さ
w	トリップの数
d	地理複合シーケンスの次元数
$\mathcal{X}$	$w \times d \times n$ 次元の地理情報テンソル：$\mathcal{X} = \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_w\}$
$\boldsymbol{X}$	d 次元の地理複合シーケンス
V-レジーム	
$\boldsymbol{m}$	$\mathcal{X}$ に含まれるセグメントの総数：$\boldsymbol{m} = \{m_1, \ldots, m_w\}$
$\mathcal{S}$	$\mathcal{X}$ に含まれるセグメント集合：$\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_w, \boldsymbol{m}\}$
r	$\mathcal{X}$ に含まれる V-レジームの総数
$\boldsymbol{\Theta}$	r 個の V-レジームのモデルパラメータ集合：$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}_{r\times r}\}$
$\boldsymbol{\theta}_i$	i 番目の V-レジームのモデルパラメータ
k_i	$\boldsymbol{\theta}_i$ の状態数
$\boldsymbol{\Delta}_{r\times r}$	V-レジーム遷移行列：$\boldsymbol{\Delta} = \{\delta_{ij}\}_{i,j=1}^{r}$
$\mathcal{V}$	セグメントメンバーシップ：$\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_w\}$
H-レジーム	
g	$\mathcal{X}$ に含まれる H-レジームの総数
$\boldsymbol{\Phi}$	g 個の H-レジームのコア集合：$\boldsymbol{\Phi} = \{\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_g\}$
$\boldsymbol{\phi}_j$	j 番目の H-レジームのコア
$\mathcal{H}$	トリップメンバーシップ：$\mathcal{H} = \{h_1, \ldots, h_w\}$
コスト関数	
$\mathcal{C}$	候補解：$\mathcal{C} = \{r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}\}$
$Cost_T(\boldsymbol{X}; \mathcal{C})$	$\mathcal{C}$ による $\mathcal{X}$ の総コスト

な課題は，(a) どのように各トリップ，各ゾーンにおける特徴を抽出するか，(b) どのようにセグメントの数，V-レジームおよび H-レジームの数を推定するか，(c) どのように 2 種類のレジームを表現し，セグメント，トリップの割当てを行うかである．本研究では，ユーザによるパラメータ設定を介せず，自動処理によって最適解を求めるための新手法を提案する．

4.　提案モデル

本章では，問題 1 を解決するためのモデルを提案する．提案モデルはモデル表現コストのアイデアに基づいており，以下に詳述する．

4.1　特徴抽出とデータ圧縮

まず，大規模センサデータを表現するため，最小記述長（MDL: minimum description length）の概念を用いる．MDL は情報理論に基づくモデル選択基準の 1 つであり，可逆圧縮を行うことができるが，そのものの概念だけでは本論文の目的を直接解決することはできない．そこで，与えられたテンソル $\mathcal{X}$ を適切に表現するモデルを見つけるために，新しい符号化スキームを導入する．

地理情報テンソル $\mathcal{X}$ が与えられたときのモデルの良さは次の式で表現できる：$Cost_T = Cost(\mathcal{M}) + Cost(\mathcal{X}|\mathcal{M})$．ここで，$Cost(\mathcal{M})$ はモデル $\mathcal{M}$ を表現するためのコストを示し，$Cost(\mathcal{X}|\mathcal{M})$ は，$\mathcal{M}$ が与えられたときの $\mathcal{X}$ の符号化のコストを示す．以下では単一のシーケンス $\boldsymbol{X}$ のコストについて議論した後，トリップ数 w の地理情報テンソル $\mathcal{X}$ のコストについて述べる．

4.2　地理複合シーケンスのモデル表現コスト

シーケンス $\boldsymbol{X}$ が与えられたとき，提案モデルの表現コストは以下の要素から構成される．

- 多次元シーケンスデータの長さ n と次元数 d: $\log^*(n) + \log^*(d)$ ビット[*4]
- セグメントと V-レジームの個数 m, r: $\log^*(m) + \log^*(r)$
- 各セグメントの V-レジームへの割当て（セグメントメンバーシップ）：$m \log(r)$ ビット
- 各セグメントの長さ s: $\sum_{i=1}^{m-1} \log^* |s_i|$ ビット
- r 個の V-レジームのモデルパラメータ集合：$Cost_M(\boldsymbol{\Theta}) = \sum_{i=1}^{r} Cost_M(\boldsymbol{\theta}_i) + Cost_M(\boldsymbol{\Delta})$．単一の V-レジームのモデル $\boldsymbol{\theta}$ は，状態数 $k(\log^*(k))$ と確率モデル ($\boldsymbol{\theta} = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{B}\}$) の表現コストが必要となる（$\boldsymbol{\pi}$ は HMM における初期確率，$\mathbf{A}$ は遷移確率，$\mathbf{B}$ は出力確率である）．まとめると，$Cost_M(\boldsymbol{\theta}) = \log^*(k) + c_F \cdot (k + k^2 + 2kd)$．ここで，$c_F$ は浮動小数点のコストを示す[*5]．同様にして，V-レジーム遷移行列には，$Cost_M(\boldsymbol{\Delta}) = c_F \cdot r^2$ のコストを要する．

4.3　地理情報テンソルの符号化コスト

先述のとおり，本論文では隠れマルコフモデルを用いてシーケンス $\boldsymbol{X}$ の車両走行パターンを表現するが，ここで重要なのは，推定したモデルが $\boldsymbol{X}$ を正しく表現しているかを判断する指標の導入である．ハフマン符号を用いた情報圧縮では，モデル $\boldsymbol{\theta}$ が与えられた際の $\boldsymbol{X}$ の符号化コストを負の対数尤度を用いて次のように表現することができる．

$$Cost_C(\boldsymbol{X}|\boldsymbol{\theta}) = \log_2 \frac{1}{P(\boldsymbol{X}|\boldsymbol{\theta})} = -\ln P(\boldsymbol{X}|\boldsymbol{\theta}). \qquad (1)$$

ここで，$P(\boldsymbol{X}|\boldsymbol{\theta})$ は $\boldsymbol{X}$ の尤度を示す．シーケンス $\boldsymbol{X}$ と r 個の V-レジームのモデルパラメータ集合 $\boldsymbol{\Theta}$ が与えられたとき，データ圧縮のためのコストは次のとおりである．

$$Cost_C(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{i=1}^{m} Cost_C(\boldsymbol{X}[s_i]|\boldsymbol{\Theta}) \cdot \sum_{i=1}^{m} -\ln(\delta_{vu} \cdot (\delta_{uu})^{|s_i|-1} \cdot P(\boldsymbol{X}[s_i]|\boldsymbol{\theta}_u))$$

ここで，i と $(i-1)$ 番目のセグメントはそれぞれ u と v 番目の V-レジームに所属し，$v_i = u$, $v_{i-1} = v$, $v_0 = v_1$ とする．また，$\boldsymbol{X}[s_i]$ はセグメント s_i の部分シーケンスを示

*4　ここで，$\log^*$ は整数のユニバーサル符号長を表す：$\log^*(x) \approx \log_2(x) + \log_2 \log_2(x) + \ldots$ [17].
*5　本論文では 4×8 ビットとする．

© 2016 Information Processing Society of Japan

し，$P(\boldsymbol{X}[s_i]|\boldsymbol{\theta}_u)$ はセグメント s_i の尤度とし，$\boldsymbol{\theta}_u$ はセグメント s_i が所属する V-レジームである．

H-レジームの表現コストは以下の要素から構成される．

- トリップの数 w と H-レジームの個数 g: $\log^*(w) + \log^*(g)$ ビット
- 各トリップの H-レジームへの割当て（トリップメンバーシップ）：$w\log(g)$ ビット

4.4　符号化コスト関数

トリップ i のシーケンスを $\boldsymbol{X}_i$，セグメント数を m_i，トリップ i の j 番目のセグメントの位置を s_{ij} とするとき $(i = 1, \ldots, w)$，候補解 $\mathcal{C} = \{r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}\}$ が与えられたときの地理情報テンソル $\mathcal{X}$ の符号長を次に示す．

$$
\begin{aligned}
Cost_T(\mathcal{X};\mathcal{C}) &= Cost_T(\mathcal{X}; r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}) \\
&= \sum_{i=1}^{w} \log^*(n_i) + \log^*(d) + \sum_{i=1}^{w} \log^*(m_i) \\
&\quad + \log^*(r) + \log^*(g) + \log^*(w) + w\log(g) \\
&\quad + \sum_{i=1}^{w} m_i \log(r) + \sum_{i=1}^{w} \sum_{j=1}^{m_i-1} \log^* |s_{ij}| \\
&\quad + Cost_M(\boldsymbol{\Theta}) + \sum_{i=1}^{w} Cost_C(\boldsymbol{X}_i|\boldsymbol{\Theta}) \qquad (2)
\end{aligned}
$$

したがって本論文の次の目標は，上記のコスト関数 (2) を最小化するようなセグメント，V-レジームおよび H-レジーム集合を発見することであり，次章ではそのためのアルゴリズムについて述べる．

5.　最適化アルゴリズム

前章では，候補解 $\mathcal{C} = \{r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}\}$ が与えられたうえでテンソル $\mathcal{X}$ を表現するためのコスト関数として，式 (2) を示した．本章では，式 (2) に基づき，最適な解 $\mathcal{C}$ を発見するためのアルゴリズム TrailMarker を提案する．

5.1　TrailMarker

本研究では，前章で述べたコストモデルに基づき，セグメント，V-レジームおよび H-レジームの個数を自動的に選択する．直感的には，データの圧縮率が高ければ，そのモデルはデータに含まれるパターンをよく表現しているといえる．つまり，候補解 $\mathcal{C}$ に対し，最小記述長に基づく $\mathcal{X}$ の符号化コスト $Cost_T(\mathcal{X}; r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H})$ が最小となるとき，$\mathcal{C}$ は適切なモデルになる．

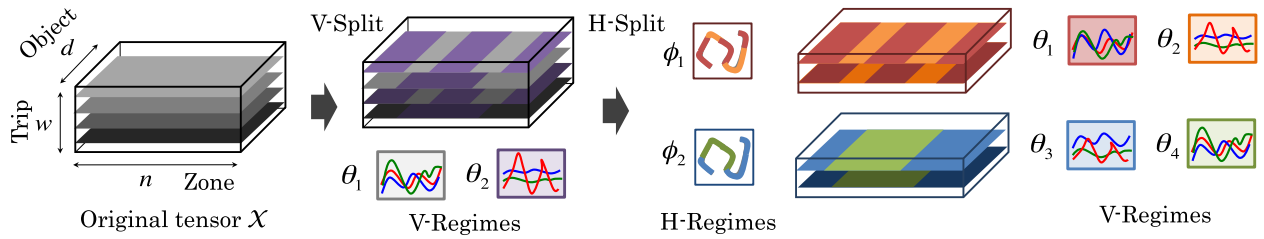
次に，具体的な最適化手法を示す．TrailMarker はスタックを用いた手法であり，貪欲法に基づく局所最適解を出力するアルゴリズムである．TrailMarker は以下に示す 2 つのステップにより，与えられた X をシーケンス方向 (vertical) とトリップ方向 (horizontal)，交互に分割する．

(1) **V-Split**：V-レジームの個数 $r = 2$ が与えられたときに，$\mathcal{X}$ をシーケンス方向 (vertical) に分割し，得られた 2 つの V-レジームを表現するモデルパラメータ $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta})$ を推定する．

(2) **H-Split**：H-レジームの個数 $g = 2$ が与えられたときに，$\mathcal{X}$ をトリップ方向 (horizontal) に分割し，得られた 2 つの H-レジームを代表するコア $(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)$ を推定する．

これにより，コスト関数である式 (2) を減少させていく．もし新しい V/H-レジーム（以下，レジームと表記）の候補のコストが現在のレジームのコストより低い場合（つまり，新しいレジームの候補ペアが勝った場合），TrailMarker は候補ペアをスタックに追加する．

図 2 は，TrailMarker の処理の流れを示している．TrailMarker は 2 種類のレジームである V-レジームと H-レジームを分割しながらテンソル $\mathcal{X}$ を適切に表現する解 $\mathcal{C}$ を発見する．オリジナルのテンソル $\mathcal{X}$ が与えられたとき（図 2 (a)），まず TrailMarker は V-Split によってテンソル $\mathcal{X}$ を 2 つの V-レジームに分割し（すなわち $g = 1$, $r = 2$），2 つのモデル $\boldsymbol{\theta}_1$ と $\boldsymbol{\theta}_2$ を推定しながらセグメンテーションを行う（図 2 (b)）．次に，図 2 (c) に示すように，H-Split では 2 つの H-レジームのコア $\boldsymbol{\phi}_1$ と $\boldsymbol{\phi}_2$ を生成する．コアは各ゾーンにおいて，$\boldsymbol{\theta}_1$ と $\boldsymbol{\theta}_2$，どちらのモデルを用いて表現されているのかを示すインデックス情報である．モデルのパラメータ（$\boldsymbol{\theta}_1$ と $\boldsymbol{\theta}_2$）とモデル選択のインデックス情報（$\boldsymbol{\phi}_1$ と $\boldsymbol{\phi}_2$）を用いながら，すべてのトリップを 2 つのグループに分割する（$g = 2, r = 4$）．そして最



(a) オリジナルテンソル $\mathcal{X}$　(b) V-Split の様子 $(g = 1, r = 2)$　(c) H-Split の様子 $(g = 2, r = 4)$

図 2　TrailMarker の概要図：TrailMarker はテンソル $\mathcal{X}$ が与えられたとき，反復処理により適切な V-レジーム/H-レジームの個数を求める

Fig. 2　Overview of the workflow of TrailMarker.

© 2016 Information Processing Society of Japan

Algorithm 1 V-Split ($\mathcal{X}$)

Input: Tensor $\mathcal{X}$
Output: (a) Number of segments assigned to each V-regime, m_1, m_2
(b) Segment sets of two V-regimes, $\mathcal{S}_1, \mathcal{S}_2$
(c) Model parameters of two V-regimes $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}\}$
Initialize models $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$;
while improving the cost **do**
  /* Find segments (phase 1) */
  $\{m_1, m_2, \mathcal{S}_1, \mathcal{S}_2\}$ = SegmentAssignment ($\mathcal{X}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}$);
  /* Update model parameters (phase 2) */
  $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}\}$ = ModelUpdate ($\mathcal{S}_1, \mathcal{S}_2$);
end while
return $\{m_1, m_2, \mathcal{S}_1, \mathcal{S}_2, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}\}$;

Algorithm 2 H-Split ($\mathcal{X}$)

Input: Tensor $\mathcal{X}$
Output: (a) Number of trips assigned to each H-regime, w_1, w_2
(b) Trip sets of two H-regimes, $\mathcal{G}_1, \mathcal{G}_2$
(c) Cores of two H-regimes, $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$
Initialize cores $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2$;
while updating *trip* sets $\mathcal{G}_1, \mathcal{G}_2$ **do**
  /* Split H-regimes (phase 1) */
  $\{w_1, w_2, \mathcal{G}_1, \mathcal{G}_2\}$ = TripAssignment ($\mathcal{X}, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2$);
  /* Update cores (phase 2) */
  $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$ = CoreUpdate ($\mathcal{X}[\mathcal{G}_1]$,$\mathcal{X}[\mathcal{G}_2]$);
end while
return $\{w_1, w_2, \mathcal{G}_1, \mathcal{G}_2, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$;

後に，2 つのグループおのおのにおいてモデルパラメータを更新する（$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4$）．

これら縦横の分割処理を交互に繰り返し，V-Split と H-Split おのおのにおいてコストが下がらなければ，レジームの分割は行わず処理を終了する．次節からは，V-Split と H-Split の詳細について述べる．

5.2 V-Split

ここで扱う問題は，V-レジームの変化点の検出とモデルパラメータの推定である．具体的には，(a) 2 つの V-レジームのモデルパラメータを推定し，同時に，(b) すべての V-レジーム変化点を検出したい．そこで本研究では，式 (2) を用いてテンソル $\mathcal{X}$ の表現コストを最小にするようなモデルパラメータの推定を行う．アルゴリズム 1 は V-Split の処理を示す．提案アルゴリズムは以下に示す 2 つのステップから構成される反復処理によって，モデルパラメータの推定を行う．

- ステップ *1*：SegmentAssignment を利用し，符号化コストが最小となる V-レジーム変化点を検出し，セグメント集合を 2 つのグループ $\{\mathcal{S}_1, \mathcal{S}_2\}$ に分割する．
- ステップ *2*：ステップ 1 で得られたセグメント集合に基づき，2 つの V-レジームのモデルパラメータ $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}\}$ を推定する．ここで，HMM のパラメータの学習には，Baum-Welch アルゴリズムを用いる．

SegmentAssignment. まず最も単純な部分問題として，テンソル $\mathcal{X}$ と，2 つの V-レジームのモデルパラメータ $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}\}$ が与えられている場合を考える．まず，SegmentAssignment は V-レジームのモデルパラメータに基づき，$\mathcal{X}$ のパターンの変化点（つまりセグメントの分割位置）の候補を検出する．続いて，モデルが与えられたうえでの符号化コスト $Cost_C(\mathcal{X}|\boldsymbol{\Theta}) = -\ln P(\mathcal{X}|\boldsymbol{\Theta})$ を最小化する，V-レジーム変化点の個数と位置を最適解として出力する．ここで重要な点として，提案アルゴリズムは高速かつ単一の走査によって，最適な V-レジーム変化点の個数と位置を検出する．ゆえに，計算時間は $O(wdnk^2)$ であるが，k はゾーンの数 n に対しきわめて小さいため無視できる．よって計算時間は $O(wdn)$ となる．

モデルパラメータの初期化. V-Split では，はじめにモデルパラメータ $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ を初期化する必要がある．最も簡易的な方法としては，地理複合シーケンス $\boldsymbol{X}$ の中に含まれる部分シーケンスをランダム抽出し，モデルの初期値に設定することである．しかし，この方法を用いる場合，初期値に大きく依存するため局所解へ収束してしまう可能性がある．そこで，本研究ではこの問題を解決するため，サンプリングに基づく手法を提案する．まず $\boldsymbol{X}$ の中から複数個のセグメント/部分シーケンスをサンプルとして均等に取り出す．次に，それぞれのサンプルセグメント s に対し，モデルパラメータ $\boldsymbol{\theta}_s$ を推定する．続いて，すべてのモデルのペア $\{\boldsymbol{\theta}_{s_1}, \boldsymbol{\theta}_{s_2}\}$ に対し，符号化コストを計算し，最も適切なペア $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ を初期モデルとして選出する．

$$\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\} = \underset{\boldsymbol{\theta}_{s_1}, \boldsymbol{\theta}_{s_2} | s_1, s_2 \in \mathcal{X}}{\arg\min} \; Cost_C(\boldsymbol{X}|\boldsymbol{\theta}_{s_1}, \boldsymbol{\theta}_{s_2}), \qquad (3)$$

ここで，$\mathcal{X} = \{s_1, s_2, \ldots\}$ は，$\boldsymbol{X}$ から取り出したサンプルの集合を示す．

ModelUpdate. HMM のモデルパラメータの推定手法である Baum-Welch アルゴリズムは，モデル $\boldsymbol{\theta}$ に対し，隠れ状態の数 k を与える必要がある．しかし，この k を手動で設定するのは非常に難しい．もし k の値を小さくすれば，データの表現能力が低くなり，適切なセグメントおよび V-レジームを求めることが困難となる．一方で，もし k を大幅に上げてしまうと，オーバフィッティングを招く．そこで本研究では，隠れ状態の個数を $k = 1, 2, 3, \ldots$ のように変化させながら，コスト関数 $Cost_M(\boldsymbol{\theta}) + Cost_C(\mathcal{X}[\mathcal{S}]|\boldsymbol{\theta})$ が最小となる k を求める．

5.3 H-Split

ここでは，V-Split と同様にテンソル $\mathcal{X}$ を 2 つの H-レジームに分割し，それらのコアを推定する．アルゴリズム 2 は H-Split の処理を示す．以下に示す 2 つのステップから構成される反復処理により，最適な H-レジームを決定

© 2016 Information Processing Society of Japan

する．

- ステップ *1*：2 つのコア $\{\phi_1, \phi_2\}$ のモデルパラメータに基づき，TripAssignment を用いて 2 つの H-レジームに分割する．
- ステップ *2*：ステップ 1 で得られた H-レジームに基づき，それぞれの H-レジームのコア $\{\phi_1, \phi_2\}$ を Core-Update により更新する．

代表トリップの初期化． H-Split において，はじめに H-レジームを代表するトリップ $\{\mathbf{\Phi}_1, \mathbf{\Phi}_2\}$ を初期化する必要がある．最も簡易的な方法としては，トリップ全体からランダムに 2 つのトリップを抽出し，代表点として設定することである．しかし，この方法では初期の代表点に大きく依存した局所解へ収束してしまう恐れがある．そこで，本研究では，尤度計算に基づき最も離れた 2 点を初期の代表トリップとする手法を用いる．まずはじめに，分割前の H-レジームのコアとそれぞれのトリップとの尤度を計算し，コアと最も異なるトリップを代表トリップとして設定する．続いて，はじめに設定した代表トリップとの尤度を計算し，最も異なる代表トリップのペア $\{\mathbf{\Phi}_1, \mathbf{\Phi}_2\}$ を決定する．

$$\mathbf{\Phi}_2 = \underset{\mathbf{\Phi}_1, \mathbf{\Phi}_2 \in \mathcal{X}}{\arg\max}\, Cost_C(\mathbf{\Phi}_1 | \mathbf{\Phi}_2) \tag{4}$$

TripAssignment. 2 つのコア $\{\phi_1, \phi_2\}$ に基づき，テンソル $\mathcal{X}$ を 2 つの H-レジームに分割する．分割する際，テンソル $\mathcal{X}$ に属する各 トリップがどちらのコアに近いかによって H-レジームを決定する．ここで，コアとの近さとは，あるトリップ i を 1 つのコア ϕ_j のモデルパラメータ（すなわち，$\mathbf{\Theta}_{\phi_j}$）で表したときの符号化コストのことである．この符号化コストがより小さくなる H-レジームにトリップ i は属するものとする．

$$h_i = \underset{j | \phi_1, \phi_2}{\arg\min}\, Cost_C(\boldsymbol{X}_i | \mathbf{\Theta}_{\phi_j}) \tag{5}$$

各トリップに対し，上記のモデルパラメータを計算するため，Algorithm 1 と同様に計算時間は $O(wdn)$ となる．

CoreUpdate. H-レジームに属するトリップが更新されると，2 つのコア $\{\phi_1, \phi_2\}$ を更新する必要がある．ここでは，説明の簡略化のため，1 つのコアのみについて説明を行う．まず，(1) H-レジーム内のトリップを 1 つ選び，(2) 選んだトリップ $\boldsymbol{X}_j$ のモデルパラメータ ($\mathbf{\Theta}_{\boldsymbol{X}_j}$) と H-レジーム内のすべてのトリップ $\{\boldsymbol{X}_i\}_{i=1}^{w}$ との符号化コストを計算する．そして，(3) その際に計算される合計コストが最小となるトリップを選び，それを新しいコアとする．

$$\phi = \underset{j | \boldsymbol{X}_j \in \mathcal{X}}{\arg\min} \sum_{i=1}^{w} Cost_C(\boldsymbol{X}_i | \mathbf{\Theta}_{\boldsymbol{X}_j}) \tag{6}$$

6. 実験

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

Q1 車両走行パターン検出に関する提案手法の有効性

Q2 パターン検出に対する計算時間の検証

Q3 レジーム抽出と変化点検出に対する精度の検証

実験は 16 GB のメモリ，Intel Core i5 3.4 GHz の CPU を搭載した OS X のマシン上で実施した．本論文では 3 つの実データ（赤坂 C，H，Y コース）を用いて検証を行う．各データは平均値と分散値で正規化（z-normalization）して使用している．

- 赤坂 C コース
  このデータセットは，**図 3** に示すコースを走行したデータである（$w = 171$，$n = 2400$）．
- 赤坂 H コース
  このデータセットは，**図 4** に示すコースを走行したデータである（$w = 13$，$n = 9100$）．
- 赤坂 Y コース
  このデータセットは，図 1 に示すコースを走行したデータである（$w = 31$，$n = 5200$）．

6.1 車両走行センサデータからの特徴抽出

図 1，図 3，図 4 は赤坂コースを走行したデータに対する車両走行パターンの検出結果を示している．センサデータとして，速度（青），左右加速度（赤），前後加速度（緑）の 3 次元から構成される値を使用している．TrailMarker は，各コースデータに対し，複数の H-レジームと V-レジーム，そして外れ走行の検出に成功している．図 3，図 4 について，H-レジーム内に複数のトリップが存在する場合，代表して 2 つのトリップの出力結果を掲載している．以下で，検出結果について考察を行う．

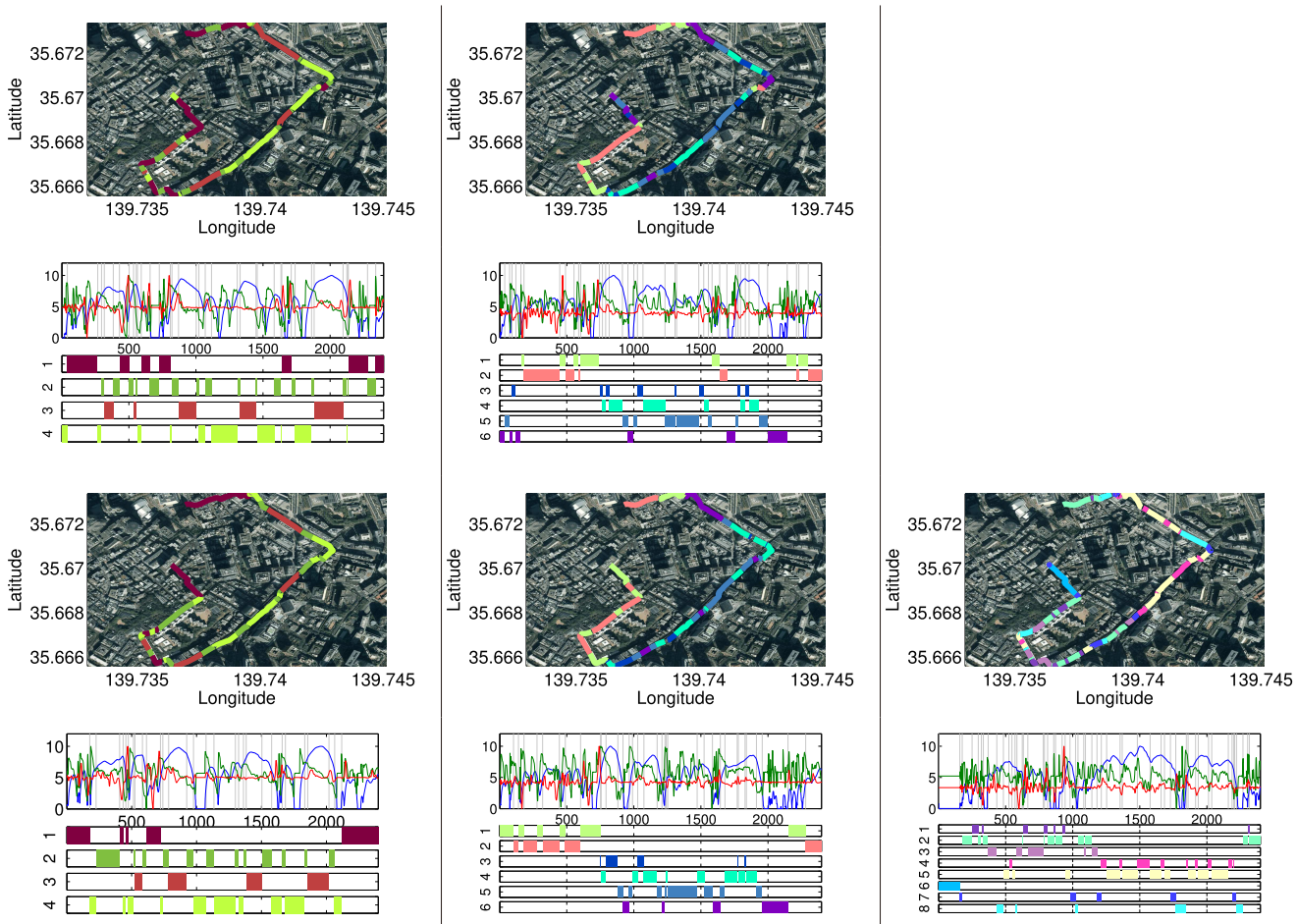
6.1.1 赤坂 Y コース

H-レジーム 1：スムーズで慣れた走行グループ． H-レジーム 1（図 1 (b)）は慣れた走行グループであり，H-レジーム 1 に属するトリップはすべて，2 回以上本コースを走行したドライバーによるものである．一時停止回数が全体平均 5.2 回に対し，この H-レジームではおよそ 3.4 回と非常に少ない．この特徴は，本コースに対するドライバーの慣れが大きく起因していると考えられ，H-レジーム 1 は他の H-レジームに比べ，安定した走行パターンが多く，いくつかの場面で一時停止することなく減速のみで対向車を回避している様子が見られる．

H-レジーム 2：慎重な走行グループ． H-レジーム 2（図 1 (a)）は慎重な走行グループである．対向車に対する減速と停止回数が最も多く，対向車を過剰に意識した慎重な走行グループであるといえる．たとえば V-レジーム 2（緑）は対向車に警戒し，減速や一時停止を行ったときに生成された V-レジームである．

外れ走行：経験の浅い走行グループ． 図 1 (c) は特に経験

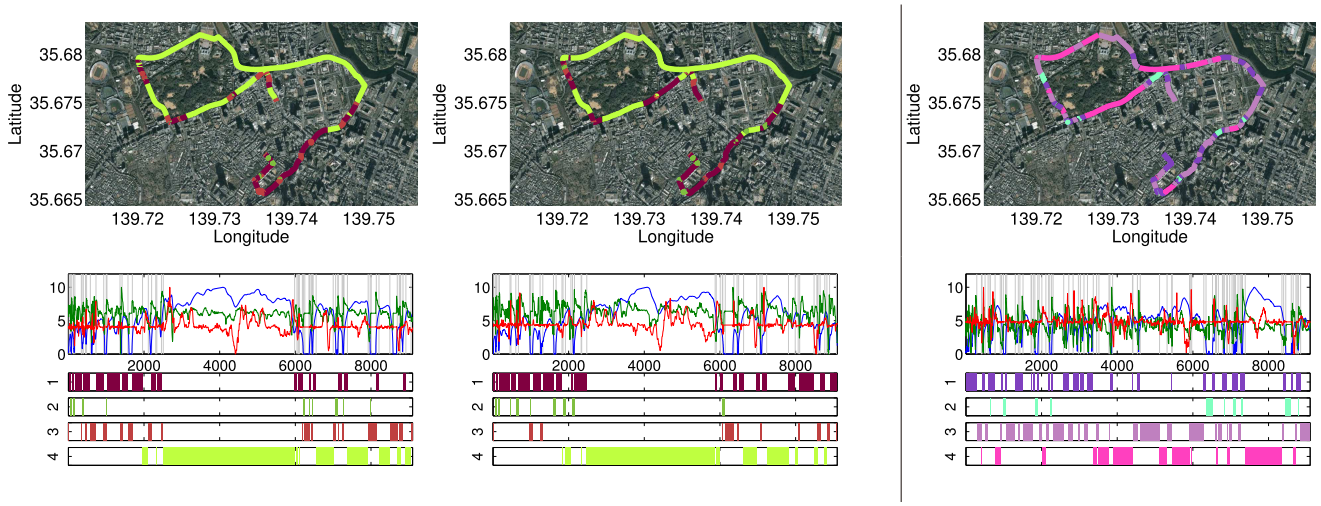
© 2016 Information Processing Society of Japan



(a) H-レジーム 1 (トリップ 数: 26/171)　(b) H-レジーム 2 (トリップ 数: 144/171)　(c) 外れ走行 (トリップ 数: 1/171)

図 3　C コースを走行したデータにおける TrailMarker の出力結果（総トリップ数：171）

Fig. 3　Result of TrailMarker (C course, 171 trips).



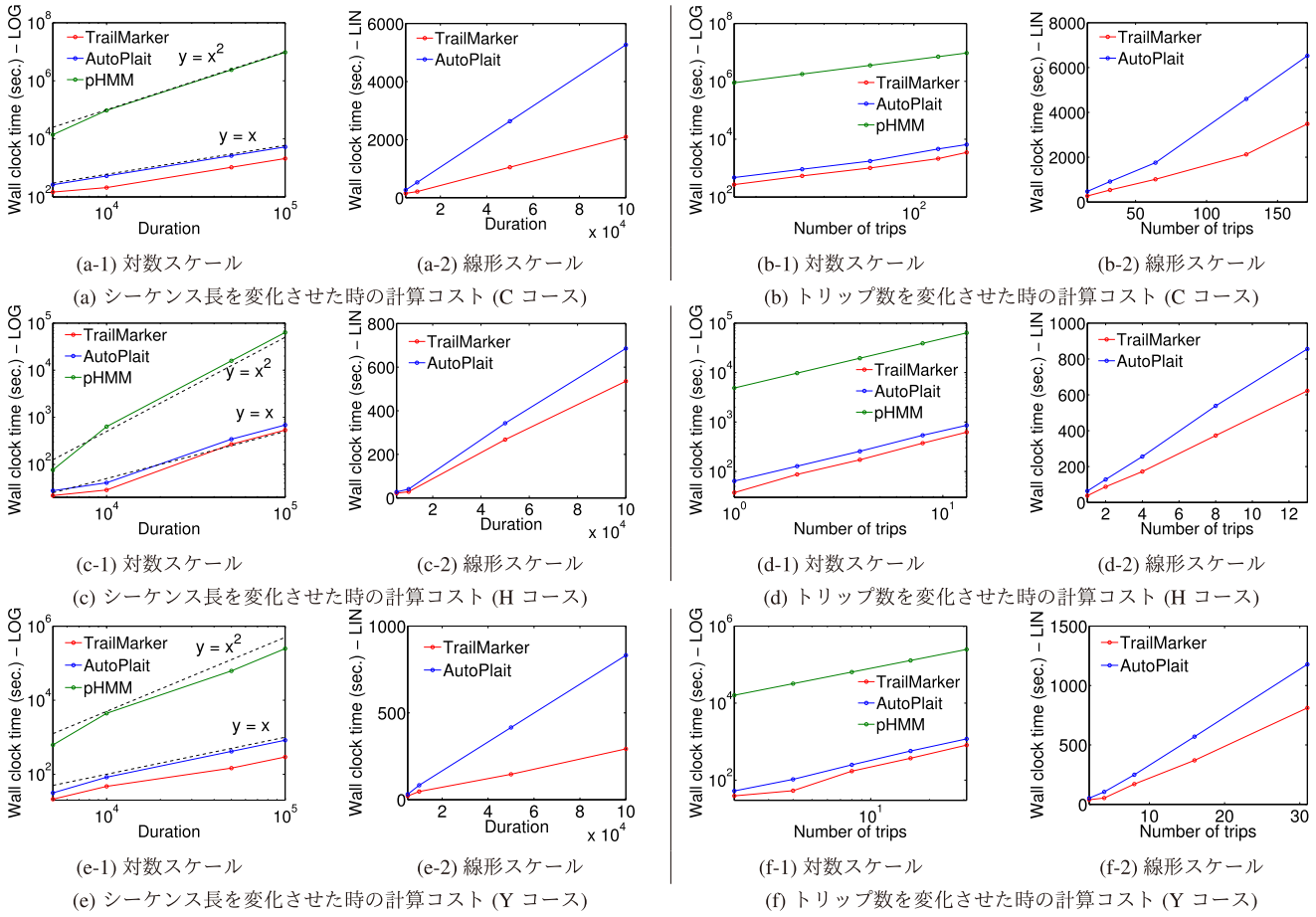
(a) H-レジーム 1 (トリップ 数: 12/13)　(b) 外れ走行 (トリップ 数: 1/13)

図 4　H コースを走行したデータにおける TrailMarker の出力結果（総トリップ数：13）

Fig. 4　Result of TrailMarker (H course, 13 trips).

の浅いトリップである．H-regime2（図 1 (a)）の特徴に加え，先行車両が存在する時間が最も長い（平均 126 秒に対し，本トリップでは 166 秒）．慣れていないドライバーである場合，交通量が少ないにもかかわらず先行車を回避することなく走行を続ける．たとえばゾーン 3100 から 3700 に見られる V-レジーム 5 と V-regime 6 は先行車を意識して加減速を繰り返したことによって生成された V-レジームである．実際に，本トリップは赤坂 Y コースを初めて走

© 2016 Information Processing Society of Japan



(a-1) 対数スケール　(a-2) 線形スケール

(a) シーケンス長を変化させた時の計算コスト (C コース)

(b-1) 対数スケール　(b-2) 線形スケール

(b) トリップ数を変化させた時の計算コスト (C コース)

(c-1) 対数スケール　(c-2) 線形スケール

(c) シーケンス長を変化させた時の計算コスト (H コース)

(d-1) 対数スケール　(d-2) 線形スケール

(d) トリップ数を変化させた時の計算コスト (H コース)

(e-1) 対数スケール　(e-2) 線形スケール

(e) シーケンス長を変化させた時の計算コスト (Y コース)

(f-1) 対数スケール　(f-2) 線形スケール

(f) トリップ数を変化させた時の計算コスト (Y コース)

図 5　TrailMarker の計算コスト

Fig. 5　Scalability of TrailMarker.

行したドライバーによるものであった．

6.1.2　赤坂 C コース

赤坂 Y コースと同様に，H-レジーム 1（図 3 (a)）はスムーズで慣れた走行グループであり，H-レジーム 2（図 3 (b)）は慎重な走行グループである．このコースは悪天候（雨，雪）時のトリップを含んでおり，雨の日のトリップのうち 87%，そして雪の日のトリップのすべてが H-regime 2 に属している．

上記に対し，図 3 (c) は外れ走行であり，特に歩行者，自転車に対する急な停止と減速が多く見られる走行である．歩行者，自転車に対する急な停止，減速回数は，C コース全体で平均して 0.7 回であるのに対し，このトリップでは 7 回の急な停止と減速が見られた．

6.1.3　赤坂 H コース

本コースでは，ゾーン 2100 から 6000 までの区間において首都高速を走行するルートを選択しているため，すべてのトリップが安定して高速な走行を行っている．結果として，すべてのトリップが 1 つの H-レジーム（図 4 (a)）に属している．ただし，図 4 (b) に示すトリップのみ，交通規制によって一般道を走行しているため，外れ走行として検出されている．

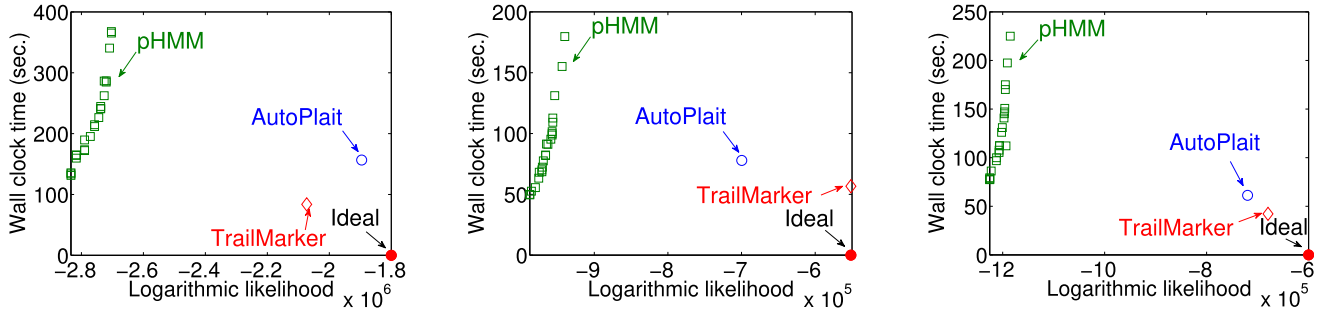
上記のように，本手法 TrailMarker はパラメータ設定や事前知識を要することなく，複雑な車両走行グループ，車両走行パターンとその変化点を発見することができる．また，これらの車両走行グループから外れた走行も自動的に検出することができる．

6.2　計算コスト

図 5 はシーケンス長 n，トリップ数 w を変化させた際の TrailMarker と比較手法における計算コストを示している．より詳細に計算コストを検証するため，対数スケールと線形スケール両方の実験結果の図を記載している．ここでは，大規模時系列データ解析の最新の手法である pHMM [22]，AutoPlait [10] と比較した．線形スケールにおいて，pHMM は計算コストが大きすぎるため，AutoPlait のみ提案手法と比較した．また，pHMM はパラメータを必要とするため，文献 [22] にしたがって $\epsilon_r = 0.1, \epsilon_c = 0.8$ とした．

TrailMarker，AutoPlait はデータの長さに対し，線形 $O(n)$ である（対数スケールにおいて傾きは $slope = 1.0$ である）．一方，pHMM は $O(n^2)$ の計算量を要する（$slope \approx 2.0$）．TrailMarker は AutoPlait と比較し，$n = 100000$ において，平均して 2.3 倍，pHMM に対しては 3366 倍の性能向上を達成している．特に，2 方向への分

© 2016 Information Processing Society of Japan



(a) C コース　(b) H コース　(c) Y コース

図 6　TrailMarker の精度と計算コスト

Fig. 6　Computation cost and accuracy of TrailMarker.

割を交互に行うことにより，より高速に解が収束するため AutoPlait よりも高い性能を示している．また，トリップ数に対しても，TrailMarker は pHMM，AutoPlait と比較し，高い性能を示している．

6.3　精度

続いて，与えられたシーケンスに対する提案手法の変化点検出とクラスタリングの精度について検証する．**図 6** は提案手法と比較手法におけるセグメントとレジーム抽出の精度と計算コストに関する実験結果である．精度については，実データに対するモデルの対数尤度に基づき評価を行う．x 軸は対数尤度を示し，実データとモデルの誤差が少ないほど大きな値となる．また，y 軸は計算コストを表すため，図 6 (a)，(b)，(c) おのおのの右下に示す赤点が理想的な結果となる．

図 6 において，TrailMarker と AutoPlait については 1 点のみで実験結果が表現されている．これは，両手法がパラメータを持たず，出力結果が 1 つに定まるためである．一方，pHMM については，モデルの学習精度に関連する閾値のパラメータとして，ϵ_r と ϵ_c の 2 つを設定しなくてはならない．本実験では，ϵ_r を 0.4 から 2.0 に変化させながら精度と計算コストを検証した．pHMM はパラメータによって精度と計算コストが大きく左右されることが分かる．

どの比較手法も，与えられたシーケンスの中から類似セグメントを検出する能力を持つが，TrailMarker は精度と計算コストの両面で優れた性能を示している．一方で，図 6 (a) においては，AutoPlait が提案手法よりも高い精度を示している．これは，AutoPlait がすべてのトリップに対して，個別にシーケンスの分割を行っているためである．C コースの走行データのように，多くの類似トリップが存在する場合，シーケンスのセグメント分割のみを行った方が精度が向上する場合がある．しかしながら，一般に収集される車両走行センサデータはさらに多くのトリップのグループを有しており，セグメント分割とモデル化のみでは一方向の特徴しかとらえることができず不十分である．また，トリップのグループ化により高速な情報要約が可能である．比較手法と異なり，提案手法はトリップのグループ化も同時に行っており，地理情報テンソル，もしくは車両走行センサデータの解析により適した手法となっている．

7.　まとめ

本論文では車両走行センサデータのための特徴自動抽出手法として TrailMarker を提案した．TrailMarker は，車両走行センサデータを地理情報テンソルとして扱い，おのおののトリップのグループ化（H-Split）と各グループにおけるトリップシーケンスのセグメンテーション（V-Split），それらを交互に行いながら複数のトリップシーケンスに共通する特徴を高速かつ自動で抽出する．様々な種類の実データを用いて実験を行い，TrailMarker の有効性を示した．

謝辞　本研究は JSPS 科研費 JP15H02705, JP16K12430, JP26730060，JP26280112，および総務省 SCOPE（受付番号 162110003）の助成を受けたものです．

参考文献

[1] Böhm, C., Faloutsos, C. and Plant, C.: Outlier-robust clustering using independent components, *SIGMOD*, pp.185–198 (2008).

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

[3] Chakrabarti, D., Papadimitriou, S., Modha, D.S. and Faloutsos, C.: Fully automatic cross-associations, *KDD*, pp.79–88 (2004).

[4] Fujiwara, Y., Sakurai, Y. and Yamamuro, M.: Spiral: Efficient and exact model identification for hidden markov models, *KDD*, pp.247–255 (2008).

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

[6] Lee, J.-G., Han, J. and Whang, K.-Y.: Trajectory clustering: A partition-and-group framework, *SIGMOD*, pp.593–604 (2007).

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

© 2016 Information Processing Society of Japan

[8] Li, L., McCann, J., Pollard, N. and Faloutsos, C.: Dynammo: Mining and summarization of coevolving sequences with missing values. *KDD* (2009).

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

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

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

[12] 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).

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

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

[15] Ng, R.T. and Han, J.: Clarans: A method for clustering objects for spatial data mining, *IEEE Trans. Knowl. Data Eng.*, Vol.14, No.5, pp.1003–1016 (2002).

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

[17] Rissanen, J.: A Universal Prior for Integers and Estimation by Minimum Description Length, *Ann. Statist.*, Vol.11, No.2, pp.416–431 (1983).

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

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

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

[21] Tatti, N. and Vreeken, J.: The long and the short of it: Summarising event sequences with serial episodes, *KDD*, pp.462–470 (2012).

[22] Wang, P., Wang, H. and Wang, W.: Finding semantics in time series, *SIGMOD Conference*, pp.385–396 (2011).

[23] Wilpon, J.G., Rabiner, L.R., Lee, C.H. and Goldman, E.R.: Automatic recognition of keywords in unconstrained speech using hidden Markov models, *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.38, No.11, pp.1870–1878 (1990).

[24] Zhang, T., Ramakrishnan, R. and Livny, M.: Birch: An efficient data clustering method for very large databases, *SIGMOD*, pp.103–114, ACM (1996).

本田 崇人

2015 年熊本大学工学部情報電気電子工学科卒業．2015 年熊本大学大学院自然科学研究科情報電気電子工学専攻博士前期課程入学．2015 年 Web とデータベースに関するフォーラム最優秀論文賞，2016 年 SIGMOD 2016 Student Travel Award 受賞．大規模センサデータマイニングに関する研究に従事．ACM，日本データベース学会各会員．

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 年熊本大学大学院自然科学研究科日本学術振興会特別研究員（PD）．2014 年より同大学院助教．この間，カーネギーメロン大学客員研究員．2016 年日本データベース学会上林奨励賞受賞．大規模時系列データマイニングに関する研究に従事．ACM，日本データベース学会各会員．

根山 亮　（正会員）

1999 年早稲田大学大学院理工学研究科修士課程修了．同年日本アイ・ビー・エム（株）入社，東京基礎研究所にて，Web サービス，分散トランザクショナルキャッシュの研究に従事．2007 年度本会喜安記念業績賞受賞．2008 年より（株）トヨタ IT 開発センターにて車両センサーデータ解析，自動運転・運転支援向けデータベースの研究に従事．2015 年 WebDB フォーラム最優秀論文賞受賞．

© 2016 Information Processing Society of Japan

櫻井 保志　（正会員）

1991 年同志社大学工学部電気工学科卒業．1991 年日本電信電話（株）入社．1999 年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了．博士（工学）．2004～2005 年カーネギーメロン大学客員研究員．2013 年熊本大学大学院自然科学研究科教授．本会平成 18 年度長尾真記念特別賞，平成 16 年度および平成 19 年度論文賞，電子情報通信学会平成 19 年度論文賞，日本データベース学会上林奨励賞，ACM KDD best paper awards（2008，2010）等受賞．データマイニング，データストリーム処理，センサーデータ処理，Web 情報解析技術の研究に従事．ACM，電子情報通信学会，日本データベース学会各会員．

（担当編集委員　宝珍 輝尚）

© 2016 Information Processing Society of Japan