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]

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，位置情報を伴う様々な車両走行センサデータが与えられたときに，各々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なうことなく，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なうことができる．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．

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な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，全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．例えば，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．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通りである．

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

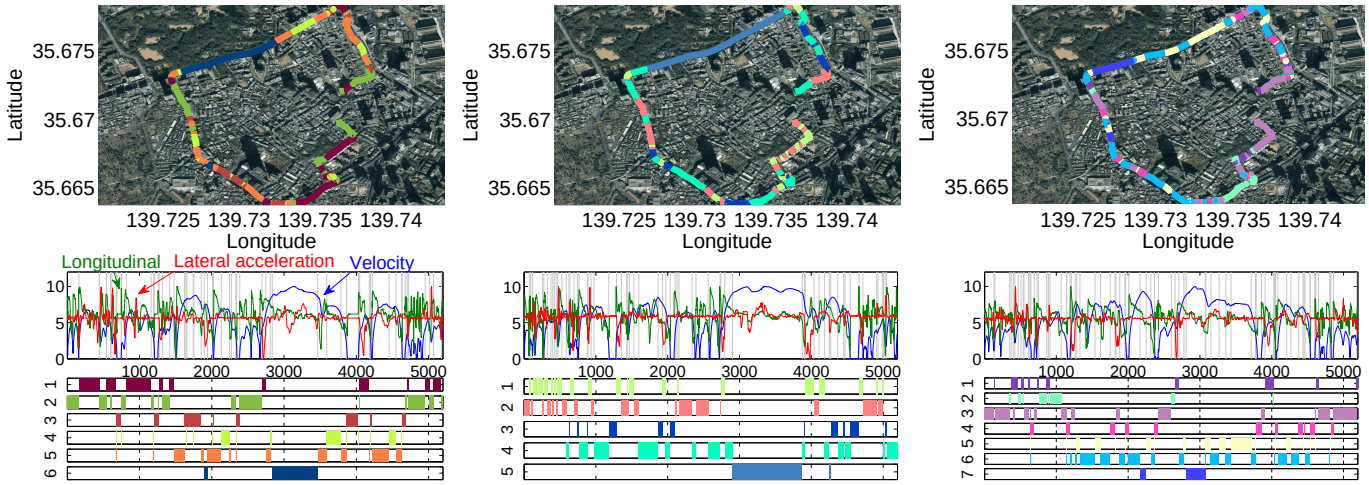
(1) $\mathcal{X}$ の中のパターンの変化点を発見し，部分シーケンス集合 (セグメント) に分割し，
(2) セグメントの共通パターンを検出するとともに，
(3) 類似した車両走行シーケンスをグループ化する．
(4) さらに重要な点として，これらの処理は高速かつ自動で行なう．

具体例．図 1 は，赤坂 Y コースの車両走行データと TrailMarker の出力結果例である．この車両走行の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) の上段は TrailMarker の出力結果を地図上にプロットしたものであり，下段は 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) を示している．図 1 (a)(b)(c) は各々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，下段には各グループにおける典型的なシーケンスをグループの代表として示している．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を抽出すると同時に，慎重な走行 (図 1 (a))，スムーズで安定した走行 (図 1 (b))，渋滞時の走行 (図 1 (c)) など車両走行のグループ化も行なうことができる．

1.1 自動抽出手法の重要性

クラスタリング [3], セグメンテーション [2], [13]，類似シーケンス探索 [9], [10] などセンサデータを対象とした研

[1] 熊本大学
[2] トヨタ IT 開発センター
[a)] takato@dm.cs.kumamoto-u.ac.jp



(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 (総トリップ 数：31).

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全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，複数の観点から全ての要素を統合的に解析する．提案手法 TrailMarker は以下の特長がある．

(1) 全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，各々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なう．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ないながら，効率的にテンソルの解析を行なう．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

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 [1], [4], [5], [7], [12]．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 [11]．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 [6], [8].

隠れマルコフモデル (HMM: Hidden Markov model) は様々な分野において，センサデータ処理手法として広く利用されている [14]．最新の研究として，Wang ら [13] は文献 [2] を改良し，pHMM (pattern-based hidden Markov model) を提案している．pHMM はセンサ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もつ．この手法は，センサ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等が必要となり，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．

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なう．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，トリップ毎に毎時刻収集される．トリップとは，特定の車両による一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，場所毎の車両走行の特徴を抽出するため，全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 箇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

[*1] 一つの 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．

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．

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．

一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 個のセグメント集合 $\boldsymbol{s}$= $\{s_1, \ldots, s_m\}$ に分割してその特徴をとらえる．m 個のそれぞれの要素はセグメントの開始点と終了点から構成され，各セグメントは重複がないものとする．そして，発見したセグメント集合を類似セグメントのグループに分類する．

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，例えば，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) 各々のトリップ 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，それら両方を行な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のみならず，$\mathcal{X}$ 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なう．

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．例えば，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．

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，全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．上記の全ての情報はコスト関数 (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 で示した通り，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 $\{r, g, \mathcal{S}, \boldsymbol{\Theta}, \boldsymbol{\Phi}, \mathcal{V}, \mathcal{H}\}$ を発見することである．ここで，この

[*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 10m 間隔に区分している．

[*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．

全パラメータ集合を候補解 $\mathcal{C}$ と呼ぶ．

定義 7 $\mathcal{X}$ を表現する全パラメータ集合 $\mathcal{C} = \{r, g, \mathcal{S}, \mathbf{\Theta}, \mathbf{\Phi}, \mathcal{V}, \mathcal{H}\}$ 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．

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 の割当てを行なうかである．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の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よ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: $Cost_M(\mathbf{\Theta}) = \sum_{i=1}^{r} Cost_M(\boldsymbol{\theta}_i) + Cost_M(\mathbf{\Delta})$. 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，$Cost_M(\mathbf{\Delta}) = c_F \cdot r^2$ のコストを要する．

[*4] ここで，$\log^*$ は整数のユニバーサル符号長を表す: $\log^*(x) \approx \log_2(x) + \log_2\log_2(x) + \ldots$.

4.3 地理情報テンソルの符号化コスト

先述の通り，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 $\mathbf{\Theta}$ が与えられたとき，データ圧縮のためのコストは次の通りである．

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

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

候補解 $\mathcal{C} = \{r, g, \mathcal{S}, \mathbf{\Theta}, \mathbf{\Phi}, \mathcal{V}, \mathcal{H}\}$ 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．

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

したがって本論文の次の目標は，上記のコスト関数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] 本論文では 4×8 ビットとする．

表 1　主な記号と定義.

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

5. 最適化アルゴリズム

続いて本章では，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．候補解 $\mathcal{C}$ に対し，$\mathcal{X}$ 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 $r = 2$ が与えられた時に，$\mathcal{X}$ 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 $g = 2$ が与えられた時に，$\mathcal{X}$ 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．

図 2 は，TRAILMARKER の処理の流れを示している．TRAILMARKER は二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．まず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 V-レジームを分割し (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なう (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，全てのトリップを 2 つのグループに分割する ($g = 2, r = 4$)．そして最後に，2 つのグループ各々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 各々において，コストが下がらなければ，レジームの分割は行な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．

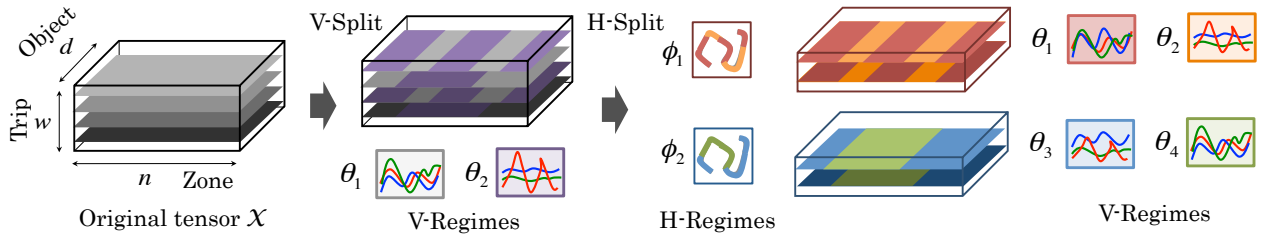
5.2 V-Split

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なう．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なう．

- ステップ *1:* CutPointSearch 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．

CutPointSearch． まず, CutPointSearch 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上での符号化コスト $Cost_C(\mathcal{X}|\boldsymbol{\Theta}) = -\ln P(\mathcal{X}|\boldsymbol{\Theta})$ 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ことができる．

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．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



(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．

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) */
  $\{m_1, m_2, \mathcal{S}_1, \mathcal{S}_2\}$ = CutPointSearch ($\mathcal{X}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}$);
  /* 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 $\{w_1, w_2, \mathcal{G}_1, \mathcal{G}_2, \boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$;

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 を決定する．

- ステップ *1:* 2 つのコア $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$ 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 $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$ を CoreUpdate により更新する．

TripAssignment．　2 つのコア $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$ 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 $\boldsymbol{\phi}_j$ のモデルパラメータ (すなわち，$\boldsymbol{\Theta}_{\boldsymbol{\phi}_j}$) で表した時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．

$$h_i = \underset{j|\boldsymbol{\phi}_1,\boldsymbol{\phi}_2}{\arg\min}\, Cost_C(\boldsymbol{X}_i|\boldsymbol{\Theta}_{\boldsymbol{\phi}_j}) \qquad (3)$$

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 $\{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\}$ を更新する必要がある．ここでは，説明の簡略化のため，一つのコア のみについて説明を行なう．まず，(1) H-レジーム 内のトリップ を一つ選び，(2) 選んだトリップ $\boldsymbol{X}_j$ のモデルパラメータ ($\boldsymbol{\Theta}_{\boldsymbol{X}_j}$) 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．

$$\boldsymbol{\phi} = \underset{j|\boldsymbol{X}_j\in\mathcal{X}}{\arg\min} \sum_{i=1}^{w} Cost_C(\boldsymbol{X}_i|\boldsymbol{\Theta}_{\boldsymbol{X}_j}) \qquad (4)$$

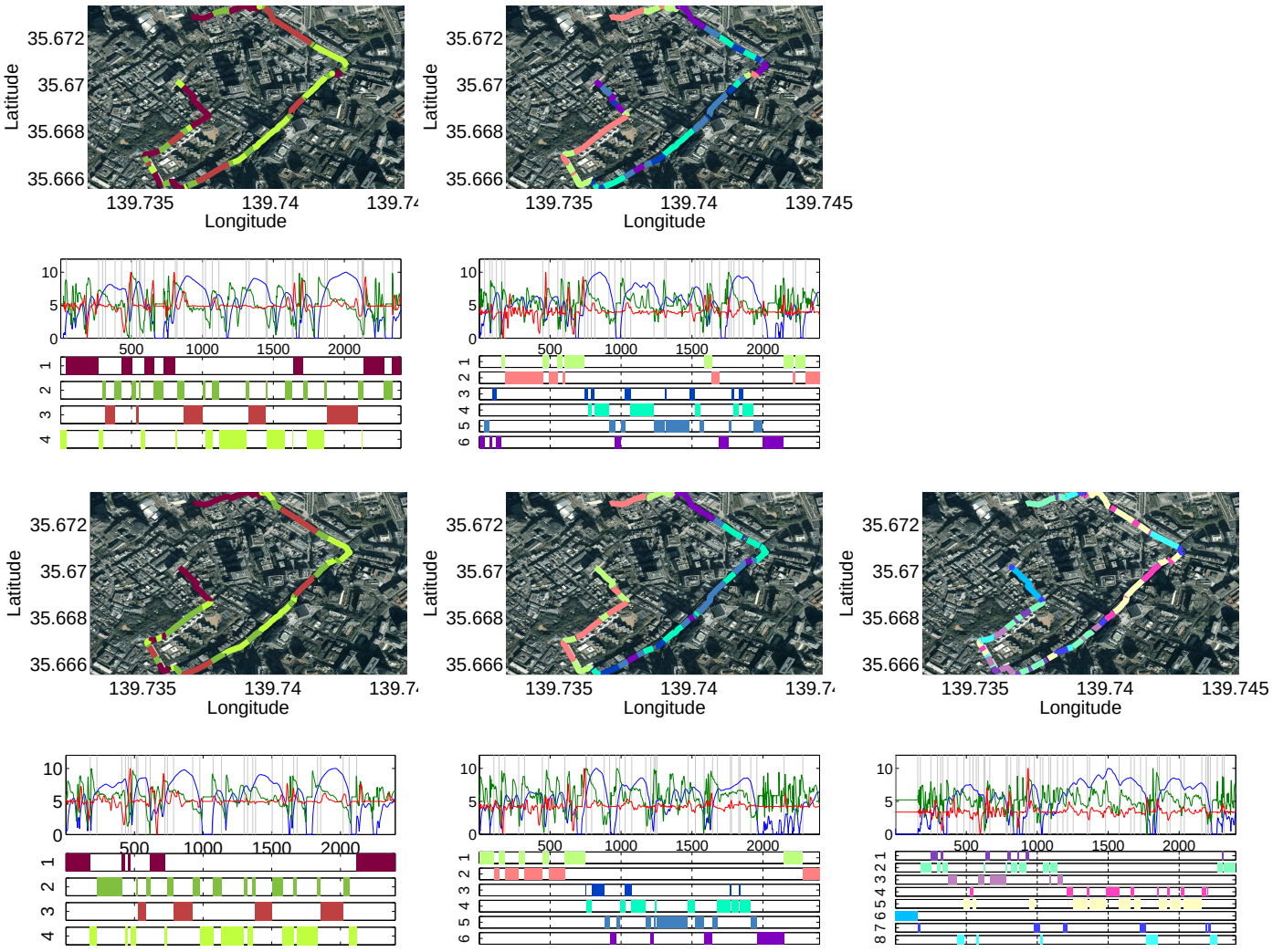
6. 実験

本論文では 3 つの実データ (赤坂 C，Y，H コース) を用いて検証を行なう．

6.1 車両走行センサデータからの特徴抽出

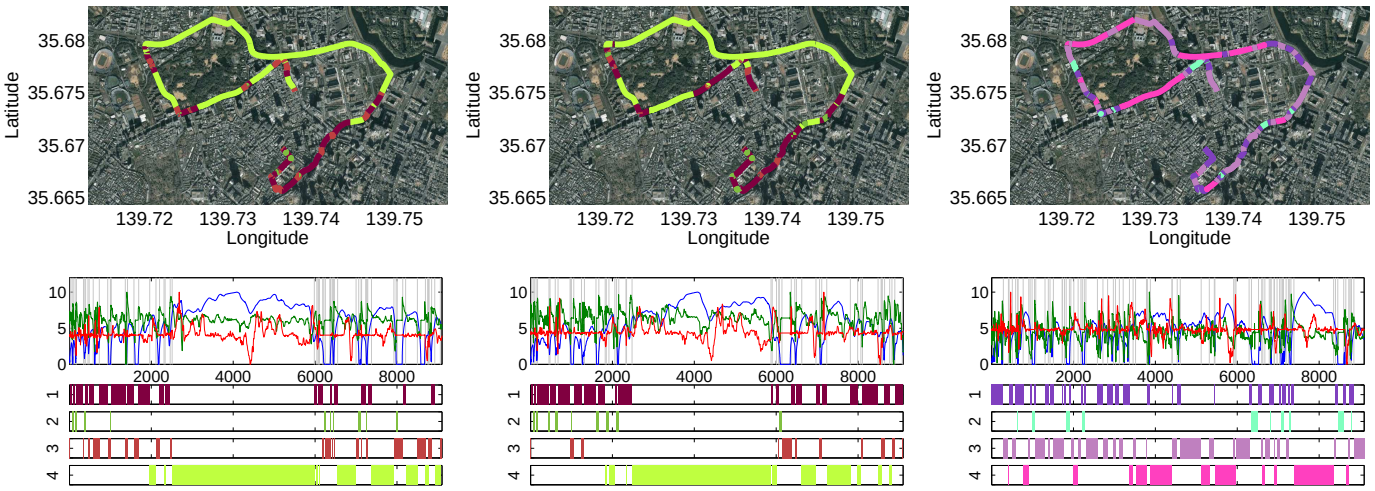
図 1，図 3，図 4 は赤坂コースを走行したデータに対する車両走行パターン抽出の結果を示している．センサデータとして，速度 (青)，左右加速度 (赤)，前後加速度 (緑) の 3 次元から構成される値を使用している．TRAILMARKER は，各コースデータに対し，複数の H-レジーム と V-レジーム，そして外れ走行の検出に成功している．以下で，検出結果について考察を行なう．

- 安定した走行 (図 1(b)，図 3(a)，図 4(a)): 道路が空いており，安定した H-レジーム である．図 4(a) のように，



(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).



(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).

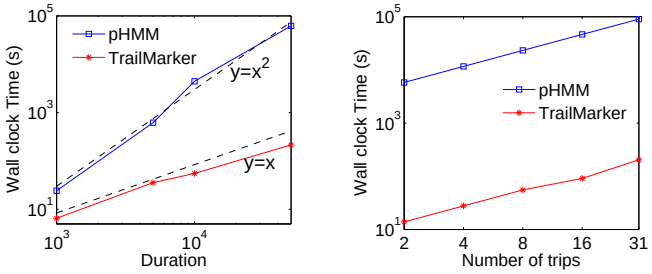


図 5　TRAILMARKER の計算コスト．

$z = 3000$〜6000 付近で速度 (青) が滑らかに増減し，前後加速度 (緑) の振動も他の H-レジーム と比較して非常に少ない．

- 渋滞時の走行 (図 1(c)，図 3(b)，図 4(b)): 図 3(b) をみると，安定した走行ができるゾーンでも，十分な速度ではない．前後加速度も激しく振動しており，先行車を意識して細かい加減速を繰り返したことがわかる．
- 慎重な走行 (図 1(a)): 図 1(a) について，$z = 3000$〜4000 に大きな直線の道路が走っているが，$z = 3500$ 付近から緩やかに左折した後，若干スピードを落として走行している．同図 (b)，(c) と比較してもこの区間の走行に，明らかな違いがあり慎重な走行をする H-レジーム であるといえる．
- 加減速の多い運転 (図 3(c)): 前後加減速の大きな増減が見られ，通常とは少々異なる走行である．これは，図 3(b) でみられた加減速とは異なり，どのようなゾーンでも現れている．また，全体として高速に走行できているにもかかわらず前後加速度の大きな増減が確認できることから，この走行状態は道の混雑によるものとは区別される．

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出来る．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発見することができる．

6.2　計算コスト

図 5 はシーケンスの長さ n，トリップの数 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．ここでは，大規模センサデータの解析のための最新の手法として pHMM [13] と比較した．pHMM はパラメータを必要とするため，$\epsilon_r = 0.1, \epsilon_c = 0.8$ とした．ここでは，余白の都合上赤坂 Y コースデータのみを用いたが，他のコースでも同様の結果が得られている．TRAILMARKER 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 pHMM と比較し，$n = 50000$ において 288 倍の性能向上を達成している．また，トリップの数に対しても，TRAILMARKER は pHMM と比較し，非常に高速に動作している．

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，各々のトリップ 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)，それらを交互に行な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ない，TRAILMARKER の有効性を示した．

参考文献

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

[2] E. J. Keogh, S. Chu, D. Hart, and M. J. Pazzani. An online algorithm for segmenting time series. In *ICDM*, pages 289–296, 2001.

[3] L. Li, B. A. Prakash, and C. Faloutsos. Parsimonious linear fingerprinting for time series. *PVLDB*, 3(1):385–396, 2010.

[4] Y. Matsubara, Y. Sakurai, and C. Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, pages 193–204, 2014.

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

[6] Y. Matsubara, Y. Sakurai, C. Faloutsos, T. Iwata, and M. Yoshikawa. Fast mining and forecasting of complex time-stamped events. In *KDD*, pages 271–279, 2012.

[7] Y. Matsubara, Y. Sakurai, B. A. Prakash, L. Li, and C. Faloutsos. Rise and fall patterns of information diffusion: model and implications. In *KDD*, pages 6–14, 2012.

[8] Y. Matsubara, Y. Sakurai, W. G. van Panhuis, and C. Faloutsos. FUNNEL: automatic mining of spatially coevolving epidemics. In *KDD*, pages 105–114, 2014.

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

[10] Y. Sakurai, C. Faloutsos, and M. Yamamuro. Stream monitoring under the time warping distance. In *ICDE*, pages 1046–1055, Istanbul, Turkey, April 2007.

[11] Y. Sakurai, Y. Matsubara, and C. Faloutsos. Mining and forecasting of big time-series data. In *SIGMOD*, pages 919–922, 2015.

[12] Y. Sakurai, S. Papadimitriou, and C. Faloutsos. Braid: Stream mining through group lag correlations. In *SIGMOD*, pages 599–610, 2005.

[13] P. Wang, H. Wang, and W. Wang. Finding semantics in time series. In *SIGMOD Conference*, pages 385–396, 2011.

[14] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(11):1870–1878, 1990.