DEIM Forum 2016 G1-6

自動パターン検出のためのストリームアルゴリズム

川畑　光希†　　松原　靖子††　　櫻井　保志††

† 熊本大学工学部情報電気電子工学科
†† 熊本大学大学院自然科学研究科情報電気電子工学専攻
E-mail: †kouki@dm.cs.kumamoto-u.ac.jp, ††{yasuko,yasushi}@cs.kumamoto-u.ac.jp

あらまし　近年，データストリーム処理に関する研究が盛んに行なわれている．本論文は，様々な時系列パターンを含むデータストリームの中から，(a) 重要な特徴を自動的に発見，それらの情報を統計的に要約し，(b) 得られた特徴に類似した部分シーケンスを検出することを目的とする．時系列ストリームは，センサデータや Web アクセス履歴等，様々なアプリケーションにおいて大量に生成されており，これらの大規模な時系列ストリームの中から典型的なパターンや異常値を発見することは非常に重要な課題である．しかし，ネットワーク分析，センサ監視等，データ量が多く緊急性が要求されるような近年のアプリケーションでは，すべてのデータを蓄積してから処理することが困難である．本研究では，このような問題を解決するアルゴリズムを提案する．実データを用いた実験では，提案手法がデータストリームから有用なパターンを正確に発見，かつそれに類似した部分シーケンスを検出できることを確認し，オンライン処理であるにもかかわらず，従来の手法と同等の精度を保ち，計算時間について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　ストリームデータ処理，特徴自動抽出

1. はじめに

時系列ストリームは，センサデータや Web アクセス履歴等，様々なアプリケーションにおいて大量に生成されている．一般に，実際に生成されるデータストリームは複数のトレンドやパターンを持つことが多い．また，ネットワーク通信のモニタリングシステムから発生するデータについては，正常と異常のパターンが考えられる．本論文では，このような時系列ストリームを対象とした自動パターン検出の手法として StreamScope を提案する．StreamScope は監視するデータストリームに関する事前知識を必要とせず，その中からユーザの直感にあった特徴を自動的に抽出し，モデルパラメータとしてその情報を圧縮する．このとき，StreamScope は刻一刻と生成されるストリームが，(1) 抽出した特徴の中のどれに該当するかをリアルタイムに把握し，(2) それらの特徴を表すモデルを最新の情報に更新しながら (3) 特徴間の変化点を発見するだけでなく，それらとは異なる (4) 新たなパターンを発見することが可能である．

より具体的には，データストリーム $\boldsymbol{X}$ が与えられたとき，(a) $\boldsymbol{X}$ 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，(b) それらのセグメントをグループ化し，類似時系列パターン（本論文では「レジーム (regime)」と呼ぶ）を発見する．さらに重要な点として，これらの処理は (c) 高速かつ，リアルタイムで行う．

1.1 自動特徴抽出とストリーム処理の重要性

時系列データを対象とした研究課題は，数多く存在する．パターン発見 [20], [14], [19], [24]，情報要約 [5], [12], [13]，クラスタリング [10]，セグメンテーション [9], [25] や類似シーケンス探索 [6], [17], [21] 等は重要な課題である．しかし，これらの先行研究はすべてのデータを蓄積してから処理することが前提となっている．したがって，すべてのデータをメモリ空間に格納することなく，高速に処理することができるストリームアルゴリズムは非常に重要である．

1.2 本論文の貢献

StreamScope は以下の特長がある．

（1） データストリームから時系列パターン（レジーム）の個数と種類をオンライン処理によって把握し，それぞれの適切な変化点をリアルタイムに発見する．さらに StreamScope は，パターン変化点の最適解の検出を保証する．

（2） 4. 章で述べるモデルにより，ユーザの直感に合致した時系列パターンの抽出を行う．

（3） StreamScope は一度検出した時系列パターンの部分シーケンスに該当するデータを必要とせず，適切なレジームの数，変化点の数を，自動的に発見することができる．

（4） 従来のオフライン手法と比較して，クラスタリングの精度を落とさず計算時間について性能向上を達成した．

2. 関連研究

関連研究は以下の 3 つに分類される

パターン検出. 時系列データの解析に関する研究は様々な分野で進められている. [1], [3], [16]. 自己回帰モデル (AR: autoregressive model), 線形動的システム (LDS: linear dynamical systems), カルマンフィルタ (KF: Kalman filters) は代表的な技術であり，これらに基づく時系列の解析と予測手法が数多く提案されている [10], [22]. 著者らの先行研究 [12], [13], [19], [20] では，主にセンサデータや Web 関連のデータを用いて，大規模時系列データ集合のための探索，モデル推定，予測手法を提案している．時系列データからの情報抽出に関しては，Li らが文献 [9] 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．また，データストリームにおけるパターン検出の研究としては，L^p 距離に基づくストリームのトレンド検出や相関検出，予測を始めとする様々な手法が提案されている． SPIRIT はデータストリームから相関とトレンドに相当する隠れ値を検出する問題に取り組んだものである [16]. BRAID はデータストリーム間の遅延相関を検出するための手法である [20].

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，時系列解析手法として広く利用されている．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 [6] 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．Wang ら [25] は文献 [7] を改良し，pHMM (pattern-based hidden Markov model) を提案している．pHMM は時系列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，時系列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．階層的確率モデルとして，Fine ら [4] は階層的 HMM (HHMM: hierarchical HMM) を提案し，Fox ら [5] はベータ過程に基づくモデルとして，BP-AR-HMM (beta process autoregressive HMM) を提案している．

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については，CLARANS [15], BIRCH [26], TRACLUS [8] を含め，様々なものが提案されている．文献 [2], [23] 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．また，著者らは文献 [11] において，大規模時系列データを対象とした自動特徴抽出手法を提案している．

3. 問題設定

ここでは本論文で必要な概念について定義を行う．データストリーム $\boldsymbol{X}$ は，$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \ldots$ の値からなる半無限長のシーケンスである．$\boldsymbol{x}_n$ は最も新しい値であり，時刻が進むごとに n は増加する．$\boldsymbol{x}_t$ は時刻 t における d 次元ベクトルとする．このようなデータストリーム $\boldsymbol{X}$ が与えられたとき，本研究は $\boldsymbol{X}$ を m 個のセグメント集合 $\mathcal{S}$= $\{s_1, \ldots, s_m\}$ に分割することを目的とする．s_i は i 番目のセグメントの開始点，終了点で構成され（つまり，$s_i = \{t_s, t_e\}$），各セグメントは重複がないものとする．本研究ではさらに，発見したセグメント集合を類似セグメントのグループ（レジーム: regime ）に分類する．

[**定義** 1]（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．これらグループをレジーム (regime) と呼び，それぞれの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．

[**定義** 2]（セグメントメンバーシップ） $\mathcal{F} = \{f_1, \ldots, f_m\}$ を，m 個の整数列とし，f_i を i 番目のセグメントが所属するレジームの番号とする $(1 \leqq f_i \leqq r)$.

本研究の目的は，長さ n の多次元時系列ストリームが与えられたときに，その部分シーケンスのセグメント化と分割位置の検出及び，レジームの発見を高速かつリアルタイムで行うことである．本論文で取り組む問題を以下のように定義する．

[**問題** 1] 多次元時系列ストリーム $\boldsymbol{X}$ が与えられたとき，$\boldsymbol{X}$ を表現するような以下の情報を抽出する．

（1） セグメントの総数 m と各セグメントの位置：
$\mathcal{S} = \{s_1, \ldots, s_m\}$

（2） レジームの総数 r とセグメントメンバーシップ：
$\mathcal{F} = \{f_1, \ldots, f_m\}$

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

これらの情報はコスト関数（式 (5)）を最小化するものを選ぶ．

本論文では，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．(注1) さらに，レジーム間の関係性を表現するため，レジーム遷移行列 $\boldsymbol{\Delta}_{r \times r}$ を使用する．レジーム遷移行列（定義 4）とコスト関数（式 (5)）についての詳細は，4. 章で説明する．

問題 1 で示した通り，本論文の目的は，$\boldsymbol{X}$ の特徴を抽出し，すべての時系列パターンを表現するパラメータ集合 $\{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ を発見することである．本論文では，この全パラメータ集合を**候補解** $\mathcal{C}$ と呼ぶ．

[**定義** 3] $\boldsymbol{X}$ を表現する全パラメータ集合 $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ を候補解と呼ぶ．候補解 $\mathcal{C}$ は，セグメント集合，各セグメントのレジームへの割当て，レジームを表現する確率モデル，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) どのようにセグメントおよびレジームの数を推定するか，(b) どのようにレジームを表現し，セグメントの割当てを行うかである．本研究では，リアルタイム処理によって最適解を求めるための手法を提案する．

4. 特徴抽出とデータ圧縮

本章では，問題 1 を解決するために使用するモデルについて説明する．

本研究では，複数のレジーム間の時系列パターンとその遷移を表現するために，多層的なモデルを使用する．具体的には，隠れマルコフの状態遷移をレジームにグループ化し，階層的な時系列パターンの遷移を表現する．つまり，隠れマルコフの状態 (state) に対して，レジームを上位層の状態 (super-state) と考え，遷移確率が以下のように定義される．

[**定義** 4]（レジーム遷移行列） $\boldsymbol{\Delta}_{r \times r}$ を r 個のレジーム群の遷移行列と呼ぶ．ここで，要素 $\delta_{ij} \in \boldsymbol{\Delta}$ は i 番目のレジームから j 番目のレジームへの遷移確率を示す．

行列 $\boldsymbol{\Delta}$ 内部の要素 $\delta_{i,j}$ は確率を表し，$0 \leqq \delta_{ij} \leqq 1$, $\sum_j \delta_{ij} = 1$ という条件を持つ．よって本論文で用いるモデルは r 個のレジーム集合 $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}_{r \times r}\}$ で表現され，$\boldsymbol{\theta}_i$ は i 番目のレジームのモデルパラメータを表現する．ここで，

(注1)：本論文で提案する枠組みは，HMM 以外の時系列モデルに適用することも可能である．

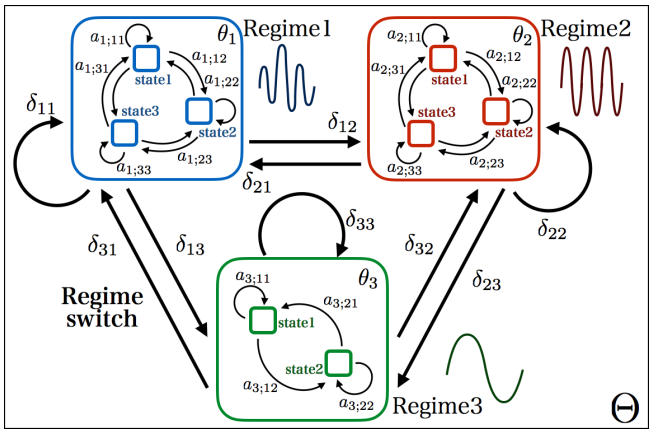


図 1　本論文で使用するモデル $\boldsymbol{\Theta}$（ここでは $r = 3$）.

$\boldsymbol{\theta}_i$ は HMM に基づき，初期確率，遷移確率，出力確率の三つ組で次のように表現される：$\boldsymbol{\theta}_i = \{\boldsymbol{\pi}_i, \mathbf{A}_i, \mathbf{B}_i\}$. (注2)

次に，最小記述長 (MDL: minimum description length) の考え方に基づき，大規模時系列データを表現する符号化スキームについて説明する．直感的には，データが与えられたときのモデルのよさは次の式で表現できる：$Cost_T = Cost(\mathcal{M}) + Cost(\boldsymbol{X}|\mathcal{M})$. 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，$Cost(\boldsymbol{X}|\mathcal{M})$ は，$\mathcal{M}$ が与えられたときのデータ $\boldsymbol{X}$ の符号化のコストを示す．表現コスト $Cost(\mathcal{M})$ 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 (注3)
- セグメントとレジームの個数 m, r：$\log^*(m) + \log^*(r)$
- 各セグメントの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 個のレジームのモデルパラメータ集合：$Cost_M(\boldsymbol{\Theta})$

$$Cost_M(\boldsymbol{\Theta}) = \sum_{i=1}^{r} Cost_M(\boldsymbol{\theta}_i) + Cost_M(\boldsymbol{\Delta}) \tag{1}$$

単一のレジームのモデル $\boldsymbol{\theta}$ 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．まとめると，

$$Cost_M(\boldsymbol{\theta}) = \log^*(k) + c_F \cdot (k + k^2 + 2kd) \tag{2}$$

ここで，c_F は浮動小数点のコストを示す．(注4) 同様にして，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．

本論文では確率モデルを用いてシーケンス $\boldsymbol{X}$ の時系列パターンを表現する．そこで，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が必要である．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．

$$Cost_C(\boldsymbol{X}|\boldsymbol{\theta}) = \log_2 \frac{1}{P(\boldsymbol{X}|\boldsymbol{\theta})} = -\ln P(\boldsymbol{X}|\boldsymbol{\theta}) \tag{3}$$

(注2)：本論文では出力確率 $\mathbf{B}$ に多次元ガウス分布を仮定する．これにより多次元ベクトルのシーケンスを確率モデルで表現する（つまり $\mathbf{B} = \{\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)\}_{i=1}^k$）.
(注3)：ここで，$\log^*$ は整数のユニバーサル符号長を表す：$\log^*(x) \approx \log_2(x) + \log_2 \log_2(x) + \ldots$. [18].
(注4)：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
シーケンス	
n	時系列の長さ
d	時系列の次元数
$\boldsymbol{X}$	d 次元の時系列シーケンス/ストリーム
$\boldsymbol{x}_t$	$\boldsymbol{X}$ の t 番目の値/要素 ($t = 1, \ldots, n$)
$\boldsymbol{X}[t_s : t_e]$	t_s から t_e までの $\boldsymbol{X}$ の部分シーケンス
セグメント	
m	$\boldsymbol{X}$ に含まれるセグメントの総数
$\mathcal{S}$	$\boldsymbol{X}$ に含まれるセグメント集合：$\mathcal{S} = \{s_1, \ldots, s_m\}$
$\mathcal{F}$	セグメントメンバーシップ：$\mathcal{F} = \{f_1, \ldots, f_m\}$
レジーム	
r	$\boldsymbol{X}$ に含まれるレジームの総数
$\boldsymbol{\Theta}$	r 個のレジームのモデルパラメータ集合：$\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}_{r\times r}\}$
$\boldsymbol{\theta}_i$	i 番目の レジームのモデルパラメータ
k_i	$\boldsymbol{\theta}_i$ の状態数
$\boldsymbol{\Delta}_{r\times r}$	レジーム遷移行列：$\boldsymbol{\Delta} = \{\delta_{ij}\}_{i,j=1}^r$
コスト関数	
$\mathcal{C}$	候補解：$\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$
$Cost_M(\boldsymbol{\Theta})$	$\boldsymbol{\Theta}$ のモデル表現コスト
$Cost_C(\boldsymbol{X}|\boldsymbol{\Theta})$	$\boldsymbol{\Theta}$ による $\boldsymbol{X}$ の符号化コスト
$Cost_T(\boldsymbol{X}; \mathcal{C})$	$\mathcal{C}$ による $\boldsymbol{X}$ の総コスト

ここで，$P(\boldsymbol{X}|\boldsymbol{\theta})$ は $\boldsymbol{X}$ の尤度を示す．

よって，シーケンス $\boldsymbol{X}$ と r 個の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の総数は次の通りである．

$$\begin{aligned} Cost_C(\boldsymbol{X}|\boldsymbol{\Theta}) &= \sum_{i=1}^{m} Cost_C(\boldsymbol{X}[s_i]|\boldsymbol{\Theta}) \\ &= \sum_{i=1}^{m} -\ln(\delta_{vu} \cdot (\delta_{uu})^{|s_i|-1} \cdot P(\boldsymbol{X}[s_i]|\boldsymbol{\theta}_u)) \end{aligned} \tag{4}$$

ここで，i と $(i-1)$ 番目のセグメントはそれぞれ u と v 番目のレジームに所属し，$f_i = u, f_{i-1} = v, f_0 = f_1$ とする．$\boldsymbol{X}[s_i]$ はセグメント s_i の部分シーケンス，$P(\boldsymbol{X}[s_i]|\boldsymbol{\theta}_u)$ はセグメント s_i の尤度，$\boldsymbol{\theta}_u$ はセグメント s_i が所属するレジームである．

まとめると，候補解 $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ が与えられたときの $\boldsymbol{X}$ の符号長は次のように表現される．

$$\begin{aligned} Cost_T(\boldsymbol{X}; \mathcal{C}) &= Cost_T(\boldsymbol{X}; m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}) \\ &= \log^*(n) + \log^*(d) + \log^*(m) + \log^*(r) + m\log(r) \\ &\quad + \sum_{i=1}^{m-1} \log^* |s_i| + Cost_M(\boldsymbol{\Theta}) + Cost_C(\boldsymbol{X}|\boldsymbol{\Theta}) \end{aligned} \tag{5}$$

本論文の最終的な目標は，上記のコスト関数を最小化するようなセグメントおよびレジーム集合を時系列ストリームの中から発見することである．

5. 最適化アルゴリズム

前章では，候補解 $\mathcal{C} = \{m, r, \mathcal{S}, \boldsymbol{\Theta}, \mathcal{F}\}$ が与えられた上でストリーム $\boldsymbol{X}$ を表現するためのコスト関数である式 (5) について述べた．続いて本章では，式 (5) に基づき，最適な解 $\mathcal{C}$ を発見するためのストリームアルゴリズムである StreamScope を提案する．

5.1 概　　要

本研究では，刻々と生成されるデータストリームから，前章で述べたコストモデルに基づき，セグメントおよびレジームの個数を自動的に選択する．直感的には，ストリームを監視しな

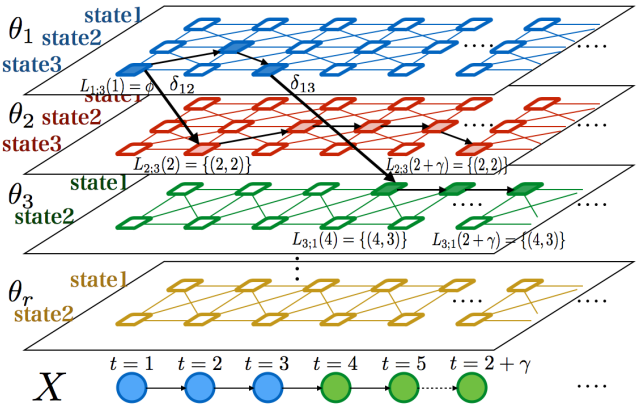


図 2　SEGMENTASSIGNMENT の様子．$\boldsymbol{X}$ と $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_r$ が与えられたとき，インクリメンタルな走査でレジームの変化点を検出し，セグメントの割り当てを行う．

がらその部分シーケンスを適切なレジームに割り当てていく．また，どのレジームにも当てはまらない場合には新たにレジームを生成する．そして，2 つの処理を繰り返しながら，時刻 n での解 $\mathcal{C}$ を最適なものに更新していく．よって，提案手法に必要な処理は以下の 2 つである．

（1）　入力 $\boldsymbol{x}_n$ に対する候補解 $\mathcal{C}$ 中のモデルとの尤度をインクリメンタルに計算し，レジーム変化点を検出する．変化点により分割したセグメントを適切なレジームに割り当てる．

（2）　候補解 $\mathcal{C}$ 中にはない新たな時系列パターンを検出し，適切なレジームを作成する．

本論文では，(1) についてのアルゴリズムである SEGMENTASSIGNMENT ついて説明し，最後に提案手法である STREAMSCOPE について説明する．

5.2　SEGMENTASSIGNMENT

ここではストリーム $\boldsymbol{X}$ からレジーム変化点を検出し，各セグメントを適切なレジームに割り当てるためのアルゴリズムを提案する．今，ストリーム $\boldsymbol{X}$ のうち，現時刻のデータであるベクトル $\boldsymbol{x}_t$ と，2 つ以上のレジームのモデルパラメータ $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}_{r\times r}\}$ が与えられていると考える．このとき，SEGMENTASSIGNMENT は レジームのモデルパラメータに基づき，$\boldsymbol{X}$ のパターンの変化点（つまりセグメントの分割位置）を検出することができる．ここで重要な点として，提案アルゴリズムは探索漏れがないことを保証しながらも，一度処理を行ったデータを遡ることなくパターン変化点を検出することができる．

図 2 は時系列ストリームからある一つのレジーム変化点を検出するときの SEGMENTASSIGNMENT の処理の様子を示している．これは青のレジーム $\boldsymbol{\theta}_1$ から緑のレジーム $\boldsymbol{\theta}_3$ へ切り替わる例である．

より具体的には，$\boldsymbol{X}$ が与えられたとき，コスト関数（式 (4)）を最小とするような，レジーム変化点をリアルタイムに発見し，それをもとに分割した部分シーケンスを 1 つのセグメント ($\mathcal{S}$) として適切なレジームに割り当てることを考える．

レジーム遷移行列 $\boldsymbol{\Delta}$ はすべてのレジームの組み合わせに対して遷移確率を持っており，本論文ではその遷移行列を用いて複数のレジーム間における変化点を検出する．そして，時刻 t における入力 $\boldsymbol{x}_t$ と，モデルが与えられた上での符号化コスト $Cost_C(\boldsymbol{X}|\boldsymbol{\Theta}) = -\ln P(\boldsymbol{X}|\boldsymbol{\Theta})$ をインクリメンタルに計算するためのアルゴリズムを提案する．

5.2.1　パターン変化点の検出

本研究では，入力 $\boldsymbol{x}_t$ と 2 つ以上のレジーム $\boldsymbol{\theta}_{1\leqq i\leqq r} = \{\boldsymbol{\pi}_i, \mathbf{A}_i, \mathbf{B}_i\}$, およびレジーム遷移行列 $\boldsymbol{\Delta}_{r\times r}$ が与えられたとき，$\boldsymbol{x}_n$ の尤度 $P(\boldsymbol{x}_n|\boldsymbol{\Theta})$ は次のようにインクリメンタルに計算される．

$$P(\boldsymbol{x}_n|\boldsymbol{\Theta}) = \max_{1\leqq i\leqq r}\{\ \max_{1\leqq u\leqq k_i}\{p_{i;u}(n)\}\quad //\ \text{regime}\ \boldsymbol{\theta}_{\mathrm{i}}\ \} \tag{6}$$

$$p_{i;u}(t) = \max\begin{cases} \delta_{ji}\cdot\max_v\{p_{j;v}(t-1)\}\cdot\pi_{i;u}\cdot b_{i;u}(\boldsymbol{x}_t) \\ \qquad //\ \text{regime switch from}\ \boldsymbol{\theta}_{\mathrm{j}}\ \text{to}\ \boldsymbol{\theta}_{\mathrm{i}} \\ \delta_{ii}\cdot\max_w\{p_{i;w}(t-1)\cdot a_{i;wu}\}\cdot b_{i;u}(\boldsymbol{x}_t) \\ \qquad //\ \text{staying at regime}\ \boldsymbol{\theta}_{\mathrm{i}} \end{cases} \tag{7}$$

$p_{i;u}(t)$ は，時刻 t におけるレジーム $\boldsymbol{\theta}_i$ の状態 u の確率の最大値を示し，式 (7) の上段は，レジーム間の切り替え（$\boldsymbol{\theta}_j$ から $\boldsymbol{\theta}_i$）の確率，下段はレジーム $\boldsymbol{\theta}_i$ の内部の状態遷移の確率を示す．また，δ_{ji} はレジーム j からレジーム i への遷移確率を示し，$max_v\{p_{j;v(t-1)}\}$ は時刻 $t-1$ における $\boldsymbol{\theta}_j$ 内の確率の最大値を示す．$\pi_{1,i}, b_{1;i}(\boldsymbol{x}_t), a_{1;ji}$ はそれぞれ $\boldsymbol{\theta}_1$ 内における状態 i の初期確率，出力確率，状態 j から状態 i への遷移確率を示す．

式 (7) より，時刻 t における遷移確率 $p_{i;u}(t)$ は，時刻 $t-1$ の遷移確率から計算される．したがって，ストリーム処理においては，時刻 $t-1$ における各レジームの各状態の遷移確率を保持しながら，累積的に尤度を計算する．そうすることで，時刻 t における尤度の計算は図 2 中の 1 つのトレリス構造についてのみであり，探索漏れがないことを保証しながらも，過去のデータを遡ることなく高速に尤度を計算することができる．

続いて，検出したレジーム変化点の候補集合について述べる．$\mathcal{L} = \{l_1, l_2, \ldots, l_i, \ldots\}$ を変化点集合とし，l_i は i 番目の変化点を表す．変化点 l は変化点の時刻と遷移先のレジーム ID から構成される．本論文では，これらの候補集合を各レジームの各状態（つまり，式 (8) において，$1\leqq i\leqq r, 1\leqq u\leqq k_i$）に対して保持し，その中から最適な変化点集合を発見する．

$$\mathcal{L}_{i;u}(t) = \begin{cases} \mathcal{L}_{j;v}(t-1)\cup\{l\} & //\ \text{switch from}\ \boldsymbol{\theta}_{\mathrm{j}}\ \text{to}\ \boldsymbol{\theta}_{\mathrm{i}} \\ \mathcal{L}_{i;u}(t-1) & //\ \text{staying at regime}\ \boldsymbol{\theta}_{\mathrm{i}} \end{cases} \tag{8}$$

ここで $\mathcal{L}_{i;u}(t)$ は時刻 t における $\boldsymbol{\theta}_i$ の状態 u の変化点集合である．これらは式 (7) の尤度計算に基づき更新される．ある時刻 t において，レジーム j からレジーム i に切り替わった場合，変化点 $l=(t,i)$ を変化点の候補として集合に加える．SEGMENTASSIGNMENT はこのような候補集合から尤度 $P(\boldsymbol{x}_n|\boldsymbol{\Theta})$ を最大化するような最適解 $\mathcal{L}_{best}$ を選出する．

[**補助定理** 1]　モデル $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}\}$ および長さ n のシーケンス $\boldsymbol{X}[1:n]$ が与えられたとき，尤度 $P(\boldsymbol{x}_n|\boldsymbol{\Theta})$ を最大にする変化点集合は最適である．

[**証明** 1]　SEGMENTASSIGNMENT は時刻 $t-1$ から時刻 t にかけての 1 つのトレリス構造について尤度を累積的に更新する．このとき，レジーム間の変化点の候補は集合としてすべて記録される．よって，各時刻において，各レジームの各状態の Viterbi パスがすべて記録されており，尤度 $P(\boldsymbol{x}_n|\boldsymbol{\Theta})$ を最大にする変化点集合は最適 Viterbi パスである．　□

［**補助定理** 2］ モデル $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}\}$ および，シーケンス $\boldsymbol{X}[1:t+\gamma]$ が与えられたとき，時刻 1 から t までの部分パスから生成されるセグメントは時刻 $t+\gamma$ において最適である．

［**証明** 2］ 尤度 $P(x_{t+\gamma}|\boldsymbol{\Theta})$ は時刻 $t+\gamma$ において確率の最大値を示す．したがって，補助定理 1 より，時刻 1 から $t+\gamma$ までの変化点集合は最適である．よって，時刻 1 から t における部分パスは時刻 $t+\gamma$ まで最適である． □

本研究では，時刻 1 から t における部分パスが補助定理 2 を満たすことを，γ 保証と呼ぶ．

［**定義** 5］ 時刻 t において，$P(x_t|\boldsymbol{\Theta})$ を出力するレジームがレジーム $\boldsymbol{\theta}_i$ への変化した場合の γ を以下のように定義する．

$$\gamma \propto \left\lceil \frac{1}{1-\delta_{ii}} \right\rceil \qquad (1 \leqq i \leqq r) \tag{9}$$

すなわち，γ は遷移先のセグメント長に基づくものである．変化点の候補を検出した場合，ただちにその変化点によりセグメントの割り当てを行わず，γ 保証された変化点集合をもとにセグメントをレジームに割り当てる．直感的には，あるレジーム $\boldsymbol{\theta}_i$ への変化点の候補を検出したとき，過去にレジーム $\boldsymbol{\theta}_i$ に割り当てられたセグメントの長さを考慮するために，γ 時刻未来において尤もらしいことが保証された変化点を用いてセグメントの割り当てを行う．

5.2.2 アルゴリズム

Algorithm 1 SEGMENTASSIGNMENT $(\boldsymbol{x}_t, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta})$

Input: Vector $\boldsymbol{x}_t$, model parameters of all regimes $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}\}$
Output: (a) Number of segments, m
(b) Segment set $\mathcal{S} = \{s_1, \ldots, s_m\}$
(c) Segment membership $\mathcal{F} = \{f_1, \ldots, f_m\}$
(d) Best cut point set $\mathcal{L}_{best} = \{l_1, l_2, \ldots\}$
/* Compute $p_{i;j}(t)$ */
for i = 1 to r **do**
  Compute $p_{i;j}(t)$ for state $j = 1, \ldots, k_i$; /* Equation (7) */
  Update $\mathcal{L}_{i;j}(t)$ for state $j = 1, \ldots, k_i$; /* Equation (8) */
end for
Choose the best cut-point set $\mathcal{L}_{best}$; /* Equation (6) */
if $|\mathcal{L}_{best}| > 0$ **and** $\gamma == 0$ **then**
  Compute γ; /* Equation (9) */
else
  if $\gamma > 0$ **then**
    $\gamma = \gamma - 1$;
  else
    /* Assign guaranteed segments into optimal regime */
    $t_s = t_c$;
    for each cut point $l = (t_e, i)$ in $\mathcal{L}_{best}$ **do**
      Create a new segment $s = \{t_s, t_e\}$;
      Add s into $\mathcal{S}$; $m = m + 1$; $f_m = i$;
      $t_s = t_e$;
    end for
    $t_c = t_e$;
    initialize $\mathcal{L}$;
  end if
end if
return $\{m, \mathcal{S}, \mathcal{F}, \mathcal{L}_{best}\}$;

アルゴリズム 1 に SEGMENTASSIGNMENT の具体的な処理を示す．SEGMENTASSIGNMENT は時刻 t における遷移確率 p，変化点集合 $\mathcal{L}$ を更新する．

次に，レジーム変化点を検出した場合，つまり，$\mathcal{L}_{best}$ に変化点候補が存在するとき ($|\mathcal{L}_{best}| > 0$)，式 (9) より γ を計算し，変化点集合が γ 保証されるのを待つ．

γ 保証が完了した場合，$\mathcal{L}_{best}$ の情報をもとに，部分シーケンスをセグメントとしてレジームに割り当てる．そして，すべての変化点集合を初期化し，カレントウィンドウの開始点 t_c に遷移したレジームの開始点を代入する．カレントウィンドウの定義については次節で説明する．

［**例** 1］ ここでは図 2 を用いてアルゴリズムの具体的な例を示す．SEGMENTASSIGNMENT は各時刻において，時刻 $t-1$ から現時刻 t についての尤度を計算する．時刻 1 において，$\boldsymbol{\theta}_1$ 内の確率 $p_{1;3}(1)$ は確率の最大値を持つ．時刻 2 において SEGMENTASSIGNMENT は $p_{1;3}(1)$ から $p_{2;3}(2)$ への変化点の候補を発見する（図では $\boldsymbol{\theta}_1$ から $\boldsymbol{\theta}_2$ への矢印）．同時に，$\mathcal{L}_{2;3}(2) = \{(2,2)\}$ を候補集合として更新する．ここで，$(2,2)$ は時刻 2 において，レジーム 2 へ遷移したことを表している．さらに，式 (9) にしたがって，γ を計算する．同様にして，2 番目の候補点として $\mathcal{L}_{3;1}(4) = \{(4,3)\}$ を保持する. 初めて変化点の候補を検出した時刻 $t = 2$ から γ 時刻後，アルゴリズムは $p_{3;1}(2+\gamma)$ が確率の最大値を持つことを明らかにし，$\mathcal{L}_{3;1}(2+\gamma) = \{(4,3)\}$ を最適解として出力する．

5.3 STREAMSCOPE

本論文の最終目標は，大規模時系列ストリームの中から，直感にあった時系列パターンを発見し，それらをグループ化することである．ここで解決すべき問題は，刻一刻と生成される時系列パターンを既知レジームにグループ化しながら，それらとは異なる新たな時系列パターンをどのようにして検出するかということである．

5.3.1 アルゴリズム

大規模時系列ストリームの中から自動的に時系列パターンを取り出す手法として STREAMSCOPE を提案する．STREAMSCOPE はコスト関数である式 (5) をもとに，与えられたストリームの中からこれまでに検出したパターンと同じパターンを検出しながら，新たな時系列パターンを検出した場合，自動的に新たなレジームを生成する．

ストリーム処理の場合，すべてのデータを蓄積して使用することができない．そこで，STREAMSCOPE では新たな時系列パターンを発見するために必要な部分シーケンスのみを保持しながら処理を行う．この部分シーケンスを本研究ではカレントウィンドウと呼ぶ．

［**定義** 6］ (カレントウィンドウ) 候補解 $\mathcal{C}$ のうち，最も新しいセグメント $s_m = \{t_s, t_e\}$ の開始点 t_s をカレントウィンドウの開始点 t_c とし，t_c から現時刻 t によって切り出されるストリーム $\boldsymbol{X}$ の部分シーケンス $\boldsymbol{X}[t_c:t]$ をカレントウィンドウとする．

アルゴリズム 2 は STREAMSCOPE の処理の流れを示している．STREAMSCOPE は毎時刻 SEGMENTASSIGNMENT によって既知のレジームによるパターンの変化点を検出する．

続いて，未知のレジーム，つまり新たな時系列パターンを

Algorithm 2 StreamScope ($\boldsymbol{x}_t$)

Input: a new vector $\boldsymbol{x}_t$ at time tick t
Output: Complete set of parameters $\mathcal{C}$, i.e.,
  (a) Number of segments, m
  (b) Number of regimes, r
  (c) Segment set $\mathcal{S} = \{s_1, \ldots, s_m\}$
  (d) Model parameters of regimes $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r; \boldsymbol{\Delta}\}$
  (e) Segment membership $\mathcal{F} = \{f_1, \ldots, f_m\}$
$\{m, \mathcal{S}, \mathcal{F}, \mathcal{L}\}$=SegmentAssignment ($\boldsymbol{x}_t, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_r, \boldsymbol{\Delta}$);
$\{m', \mathcal{S}', \boldsymbol{\theta}'\}$=RegimeGeneration($\boldsymbol{X}[t_c : t]$);
if $\mathcal{S}' \neq \phi$ **then**
  Compute $Cost_{SW}$ and $Cost_{CUT}$ /* Equation (11), (12) */
  if $Cost_{SW} > Cost_{CUT}$ **or** $|\mathcal{L}| == 0$ **then**
    $\mathcal{S} = \mathcal{S} \cup \mathcal{S}'$; $\boldsymbol{\Theta} = \boldsymbol{\Theta} \cup \boldsymbol{\theta}'$; $r = r + 1$;
    Update t_c;
    Update $\boldsymbol{\Delta}_{r \times r}$; /* Equation (13) */
    $f_i = r \quad (i = m + 1, \ldots, m')$; $m = m + m'$;
  end if
else
  $i = f_m$;
  Update $\boldsymbol{\theta}_i$;
end if

検出するため，RegimeGeneration によってカレントウィンドウを分割することを試みる．直感的には，1 つのモデルパラメータ $\boldsymbol{\theta}$ では表現できない時系列パターンを発見したとき，それは新たな時系列パターンであり，それを表現するためのモデル $\boldsymbol{\theta}$，およびレジームを新たに作成する．

RegimeGeneration. カレントウィンドウ $\boldsymbol{X}[t_c : t]$ が与えられたとき，RegimeGeneration は式 (5) を用いてカレントウィンドウ内における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．具体的には以下の 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:* 定理 1 より，カレントウィンドウの符号化コストが最小となる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 つのレジームのモデルパラメータ $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$ を推定する．HMM のパラメータの学習には，Baum-Welch アルゴリズムを用いる．分割して得られたモデルパラメータが，分割しない場合のモデルパラメータを用いたときのコスト関数 (式 (5)) をより小さくする場合，RegimeGeneration は新たなパターンを表現するレジームを作成する．

モデルパラメータの初期化. RegimeGeneration 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，カレントウィンドウ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．まずカレントウィンドウ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．

$$\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\} = \underset{\boldsymbol{\theta}_{s_1}, \boldsymbol{\theta}_{s_2} | s_1, s_2 \in \mathcal{X}}{\arg\min} \; Cost_C(\boldsymbol{X} | \boldsymbol{\theta}_{s_1}, \boldsymbol{\theta}_{s_2}) \tag{10}$$

ここで，$\mathcal{X} = \{s_1, s_2, \ldots\}$ は，カレントウィンドウから取り出したサンプルの集合を示す．

モデルパラメータの推定. 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，オーバー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 $Cost_M(\boldsymbol{\theta}) + Cost_C(\boldsymbol{X}[\mathcal{S}]|\boldsymbol{\theta})$ が最小となるような k を求める．

RegimeGeneration が処理されるとき，SegmentAssignment において変化点の候補が検出されていない，もしくは γ 保証をすでに終えている場合，新たに作られたレジームは最適であることが保証されている．なぜならば，既存のレジームの中で最も当てはまりの良いモデルと比較して，それを複数に分割し，新たなレジームを作成した方が，パターンをうまく表現できているという結果が得られるからである．

しかしながら，γ 保証が完了していない状態で RegimeGeneration により新たなレジームが発見された場合，γ 保証が完了した後に遷移したであろうレジームと，新たに作られたレジームは同じ時系列パターンを表現している可能性がある．

よって，本研究では，時刻 t における $\mathcal{L}_{best}$ の変化点によるセグメント集合に必要なコストと，RegimeGeneration によって出力されたレジームのカレントウィンドウに必要なコストを比較することで，新たに作られたレジームが解 $\mathcal{C}$ 内のレジームと重複しないことを保証する．より具体的には次の 2 つのコストを計算し，$Cost_{CUT}$ が $Cost_{SW}$ より小さくなるとき，新たなレジームを作成する．

$$Cost_{SW} = Cost_C(\boldsymbol{X}|, \boldsymbol{\Theta}_C) \tag{11}$$

$$Cost_{CUT} = Cost_C(\boldsymbol{X}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\Delta}) + Cost_M(\boldsymbol{\theta}) \tag{12}$$

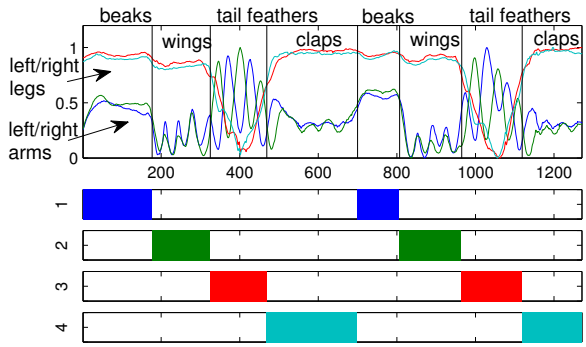
ここで，$\boldsymbol{\Theta}_C$ は，$\mathcal{L}_{best}$ に含まれるすべてのモデルパラメータの集合とする．

最後に，新たな時系列パターンが発見されず，既存のレジームに割り当てられた場合，割り当てられたデータをもとに，パラメータ $\boldsymbol{\theta}$ を更新する．

レジーム遷移確率 $\boldsymbol{\Delta}$ の更新. 未知レジームの発見により候補解 $\mathcal{C}$ 中のレジーム数 r が変更された場合，レジーム遷移確率 $\boldsymbol{\Delta}$ を以下の式によって更新する．

$$\delta_{ii} = \frac{\sum_{s \in \mathcal{S}_i} |s| - \sum_{j=1}^{r} N_{ij}}{\sum_{s \in \mathcal{S}_i} |s|}, \quad \delta_{ij} = \frac{N_{ij}}{\sum_{s \in \mathcal{S}_i} |s|} \tag{13}$$

ここで $\sum_{s \in \mathcal{S}_i} |s|$ はレジーム $\boldsymbol{\theta}_i$ に所属するセグメントの長さの総和を示し，N_{ij} は $\boldsymbol{\theta}_i$ から $\boldsymbol{\theta}_j$ へのレジームの切替回数を示す．



(a) StreamScope の出力結果

(b) 「チキンダンス (chicken dance)」における 4 つの代表的なステップ

図 3　*MoCap* データにおける StreamScope の出力結果.

6. 評価実験

本論文では StreamScope の有効性を検証するため，実データを用いた実験を行なった．実験は 16GB のメモリ，Intel Core i5 3.3GHz の CPU を搭載した iMac 上で実施した．実験で用いたデータは以下の通りであり，各々は平均値と分散値で正規化 (z-normalization) して使用した．

- *MoCap*: *MoCap* は，1 秒 120 フレームでヒトの動きを計測したモーションキャプチャのデータセットである．(注5) 本実験ではデータの中から左右の腕と足の 4 次元から構成される加速度の値を使用した．
- *GoogleTrend*(注6): このデータセットは，Google による検索クエリの頻度を週毎に 12 年間に渡り集計したものである．各シーケンスは各クエリの出現頻度を表す．

6.1 時系列データからの特徴抽出

図 3 (a) は，*MoCap* データにおける「チキンダンス (chicken dance)」(注7) の時系列シーケンスデータと，StreamScope の出力結果である．このモーションは，4 次元のシーケンスで構成され，それぞれの次元が，左右の腕と足の加速度を表現している．チキンダンスは，図 3 (b) に示す通り，beaks, wings, tail feathers, claps の 4 つの代表的なステップから構成される．図 3 の下の段は，StreamScope が自動抽出した 4 つのレジームを示している．ストリーム処理を進めるうち，StreamScope は “beak”，“wing” と次々にそれぞれのステップの特徴をレジームとして抽出し，4 つすべての特徴を捉えることに成功している．さらに，StreamScope は “clap” から再び “beak” にステップが変化したことを捉えていることから，過去に抽出した特徴を表すモデルが正しく学習できていること，そしてそれに類似するパターンを検出することができたと言える．

(注5)：http://mocap.cs.cmu.edu/

(注6)：http://www.google.com/insights/search/

(注7)：Chicken dance: http://www.youtube.com/watch?v=6UV3kRV46Zs&t=49s

6.2 計算コスト

本研究では StreamScope の性能を確認するために実際的な状況で実験を行った．図 4 は StreamScope と，時系列シーケンスから特徴抽出するためのオフライン手法である AutoPlait [11]，pHMM [25] との計算時間の比較である．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．データ集合には *MoCap* を用い，シーケンス長 n を変化させている．

StreamScope の処理はカレントウィンドウ内のデータのみを必要とする．そのため，StreamScope の計算時間はシーケンス長 n に依存することなく，すべてのデータを必要とするオフライン処理に比べ高速に動作する．StreamScope はオフライン手法である AutoPlait と比較して 100 倍以上の性能向上を達成した．

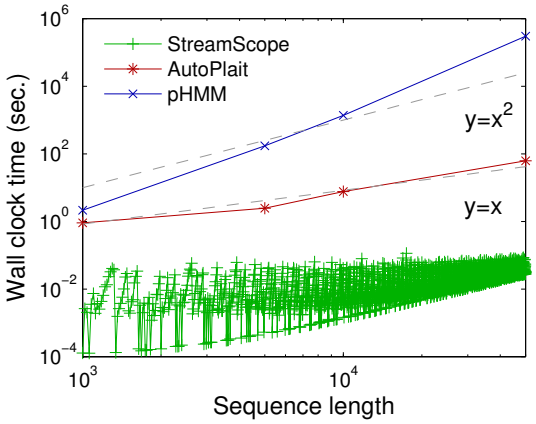


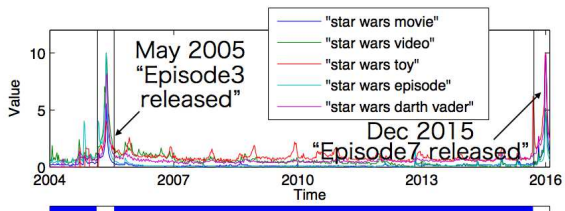
図 4　StreamScope の計算コスト.

7. アプリケーション

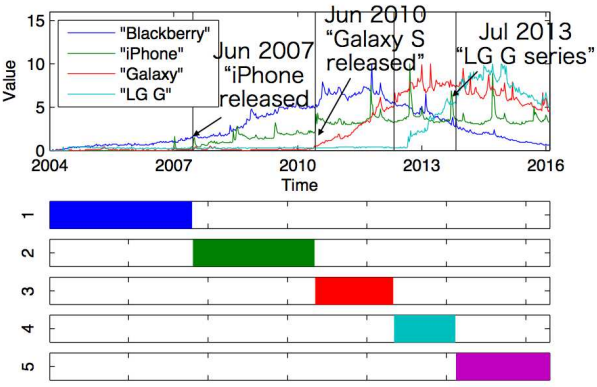
本章では，StreamScope の実用的なアプリケーションとして，*GoogleTrend* データを使用して，Web 上のユーザ動向の情報を自動検出する例を紹介する．StreamScope は時系列データの中から，未知のパターンと任意の数のレジームを自動的に発見することができる．また，さらに解析を進めることで，一度発見したパターンと類似したパターンを発見するこも可能である．

外れ値検出. 図 5 (a) は，スターウォーズに関連するキーワード 5 つ（“star wars movie”, “star wars video” 等）の 12 年間の検索数を示している．このデータは毎年 12 月に検索数が増加するという周期性を持っている．しかし，それ以外に，映画の公開によって検索数が爆発的に増加するという特徴を持っている．はじめに，StreamScope は 2005 年に映画が公開されたことを例外的なパターン (レジーム#2) として検出することに成功している．その後，通常時の周期性 (レジーム#1) に戻り，2015 年に映画が公開されたとき，過去に映画が公開されたときと同様のパターンとして，レジーム#2 への変化を検出することに成功している．

トレンド発見. 図 5 (b) はスマートフォンに関連するキーワード（“iphone”, “galaxy” 等）の時系列データであり，それぞれの



(a) スターウォーズ関連ワード （レジーム $r = 2$）



(b) スマートフォン関連ワード （レジーム $r = 5$）

図 5　*GoogleTrend* データにおけるトレンドの変化点抽出例.

データが新機種の発表や発売，または 12 月に検索数が増加する複雑なシーケンスになっている．StreamScope は過去 12 年間のスマートフォンのシェア拡大の様子を 5 つのレジームとしてそれぞれリアルタイムに検出した．具体的には，(1) スマートフォンという言葉はまだ普及しておらず，欧米で blackberry がビジネスマンの間で流行し始めた時代．(2)iPhone が発売され，スマートフォンが注目を浴び始めた時代．(3)Android 搭載端末としての Galaxy シリーズのシェア拡大期．そして，(4)LG エレクトロニクスのスマートフォンが普及し始め，(5)iOS 端末の iPhone，そして Android 端末の Galaxy，LG スマートフォンが世界シェアのほぼ全体をしめる今日の傾向といった 5 つの特徴を捉えている．

8. む　す　び

本論文では自動パターン検出のためのストリームアルゴリズムとして StreamScope を提案した．StreamScope は，与えられた時系列ストリームに対し，ユーザの直感に合う複雑な時系列パターン（レジーム）とその変化点を発見することができる．様々な種類の実データを用いて実験を行い，StreamScope はオンライン処理でありながら，その有用性を確認することができた．また，計算時間についても，従来の手法と比較し大幅に向上していることを示した．

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] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully automatic cross-associations. In *KDD*, pages 79–88, 2004.

[3] L. Chen and R. T. Ng. On the marriage of lp-norms and edit distance. In *VLDB*, pages 792–803, 2004.

[4] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[5] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky. Sharing features among dynamical systems with beta processes. In *NIPS*, pages 549–557, 2009.

[6] Y. Fujiwara, Y. Sakurai, and M. Yamamuro. Spiral: efficient and exact model identification for hidden markov models. In *KDD*, pages 247–255, 2008.

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

[8] J.-G. Lee, J. Han, and K.-Y. Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD Conference*, pages 593–604, 2007.

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

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

[11] Y. Matsubara, Y. Sakurai, and C. Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, 2014 (to appear, http://www.cs.kumamoto-u.ac.jp/~yasuko/PUBLICATIONS/sigmod14-autoplait.pdf).

[12] 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.

[13] 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.

[14] A. Mueen and E. J. Keogh. Online discovery and maintenance of time series motifs. In *KDD*, pages 1089–1098, 2010.

[15] R. T. Ng and J. Han. Clarans: A method for clustering objects for spatial data mining. *IEEE Trans. Knowl. Data Eng.*, 14(5):1003–1016, 2002.

[16] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *Proceedings of VLDB*, pages 697–708, Trondheim, Norway, August-September 2005.

[17] 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.

[18] J. Rissanen. A Universal Prior for Integers and Estimation by Minimum Description Length. *Ann. of Statist.*, 11(2):416–431, 1983.

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

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

[21] Y. Sakurai, M. Yoshikawa, and C. Faloutsos. Ftw: Fast similarity search under the time warping distance. In *PODS*, pages 326–337, 2005.

[22] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of ACM SIGMOD*, pages 611–622, 2004.

[23] N. Tatti and J. Vreeken. The long and the short of it: summarising event sequences with serial episodes. In *KDD*, pages 462–470, 2012.

[24] M. Toyoda, Y. Sakurai, and Y. Ishikawa. Pattern discovery in data streams under the time warping distance. *VLDB J.*, 22(3):295–318, 2013.

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

[26] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: an efficient data clustering method for very large databases. In *SIGMOD*, pages 103–114. ACM, 1996.