# 多次元ストリーミング時系列データの効率的なモチーフモニタリングアルゴリズム

加藤 慎也[1,a)]　天方 大地[1,b)]　原 隆浩[1,c)]

**概要**：近年，多くの IoT 機器は多次元ストリーミング時系列データを生成しており，それらを分析することに注目が集まっている．時系列データを分析する最も重要な技術として，時系列データの中に最も多く現れるサブシーケンスであるレンジモチーフがある．本稿では，多次元ストリーミング時系列データに対してレンジモチーフをモニタリングする問題に取り組む．この問題を解決するため，新たな値を観測した際，新たに生成された多次元サブシーケンスとこれまでに生成された全ての多次元サブシーケンスとの距離を計算することが考えられるが，これは効率的ではない．そのため，効率的にレンジモチーフをモニタリングするアルゴリズム MMM（Multidimensional Motif Monitoring）を提案する．MMM では，サブシーケンスをクラスタに分割し，三角不等式を用いることで不必要な距離計算の回数を削減する．4 つの実データを用いた実験により，MMM の有効性を確認する．

## 1. 序論

近年，多くのストリーミング時系列データが生成されており，それらを分析することに注目が集まっている．モチーフ発見は時系列データを分析する最も重要な技術の 1 つである [14], [20]．ある時系列データ $t$ が与えられたとき，$t$ のレンジモチーフとは，$t$ の中で最も多く現れるサブシーケンスである [15]．つまり，レンジモチーフは頻繁に発生するサブシーケンスを表す．レンジモチーフを発見することで，根底にある事象を理解したり，時系列データの特徴を知ることができる．

また，1 次元の時系列データだけでなく多次元の時系列データも多く生成されている [3], [4], [13], [16], [17], [19]．例えば，IoT 機器は複数のセンサを搭載しており，また，加速度センサやジャイロセンサは 3 方向に対する値を観測しているため，多次元時系列データを生成している．そのため，多くの研究で多次元時系列データのモチーフを発見する問題に取り組んでいる．それらの研究の多くは，多次元時系列データの全ての次元を用いてモチーフを発見している．一方，最新の研究 [19] では，一部の次元のみを用いることで，より有益なモチーフを発見できると主張している．本稿ではこの観測に基づき，多次元ストリーミング時系列データにおけるレンジモチーフ（多次元レンジモチーフ）をモニタリングする問題に取り組む．今後，特に明記する必要がない場合，多次元レンジモチーフを単にモチーフと呼ぶ．

**アプリケーション例．** IoT 機器が定期的にデータを収集し，サーバに送信すると仮定する．IoT 機器は複数のセンサを搭載しており，また，高頻度にデータを収集する．全てのデータをサーバに送信すると，多くの通信量がかかり，それらのデータを保存するために多くのストレージ容量が必要となる．そこで，得られたモチーフのみを送信および保存することで，通信量およびストレージ容量を削減できる．また，IoT 機器の管理者が時系列データをモニタリングすると仮定する．このとき，モチーフをモニタリングすることで，モチーフの変化から様々な潜在的事象を分析できる．

**提案アルゴリズムの概要．** 上記のようなアプリケーションでは，時々刻々と値が追加される多次元ストリーミング時系列データのモチーフをリアルタイムにモニタリングする必要がある．そのため，モチーフを効率的にモニタリングするアルゴリズム MMM（**M**ultidimensional **M**otif **M**onitoring）を提案する．新たな値を取得したとき，新たな値を含む新たな多次元サブシーケンス $\boldsymbol{s_n}$ が生成される．このとき，モチーフを更新する最も単純な手法として，$\boldsymbol{s_n}$ とこれまでに生成された全ての多次元サブシーケンスとの距離を計算するものが考えられる．この手法は，$\boldsymbol{s_n}$ と類似

[1] 大阪大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[a)] kato.shinya@ist.osaka-u.ac.jp
[b)] amagata.daichi@ist.osaka-u.ac.jp
[c)] hara@ist.osaka-u.ac.jp

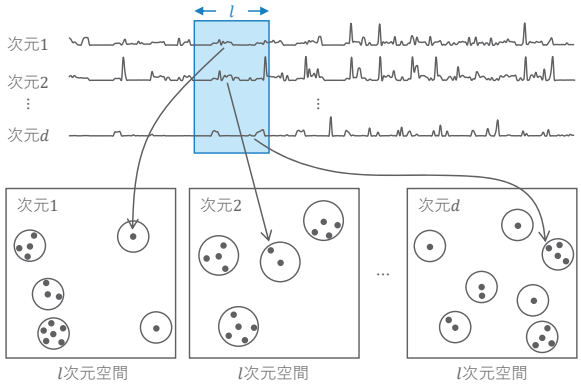



図 1: 提案アルゴリズム MMM

する多次元サブシーケンスの数を正確に取得できるが，多大な計算コストがかかる．そこで，MMM は，各次元の全てのサブシーケンスを，あるサブシーケンスを中心とするクラスタに分割する（図 1）．ある次元 $i$ に新たなサブシーケンス $s_n^{(i)}$ が生成されたとき，$s_n^{(i)}$，あるクラスタの中心サブシーケンス，およびそのクラスタ内のサブシーケンスに対して三角不等式を適用することで，そのクラスタに含まれるあるサブシーケンスと $s_n^{(i)}$ との距離の下界値を高速に計算できる．これにより，$s_n^{(i)}$ と類似しないサブシーケンスの距離計算を削減できる．

**貢献．** 以下に本研究の貢献を示す．

- 多次元ストリーミング時系列データのレンジモチーフをモニタリングする問題に取り組む．筆者らの知る限り，この問題はこれまでに取り組まれていない．
- 値を取得したときに，モチーフを効率的に更新するアルゴリズム MMM を提案する．
- 4 つの実データを用いた実験により，MMM の有効性を確認する．

**本稿の構成．** 2 章で本稿の問題を定義し，3 章で関連研究について述べる．4 章で MMM について説明し，5 章で実データを用いた実験の結果を示す．最後に 6 章で本稿のまとめと今後の課題について述べる．

## 2. 予備知識

### 2.1 定義

まず，1 次元のストリーミング時系列データを定義する．

**定義 1（ストリーミング時系列データ）．** ストリーミング時系列データ $t$ は実数値の系列であり，$t = (t[1],\ t[2],\ \dots)$ と表現する．

次に，$t$ の一部を表すサブシーケンスを定義する．

**定義 2（サブシーケンス）．** $t$ および長さ $l$ が与えられたとき，$p$ 番目の値を始点とする $t$ のサブシーケンス $s_p$ は式 (1) により定義される．

$$s_p = (t[p], t[p+1], \dots, t[p+l-1]) \tag{1}$$

ここで，$s_p$ の $x$ 番目のデータの値を $s_p[x]$ と表現する．つまり，$s_p = (s_p[1], s_p[2], \cdots, s_p[l])$ である．次に，時系列データ間の距離を測る基本的な指標である $z$ 正規化ユークリッド距離を定義する [12]．

**定義 3（$z$ 正規化ユークリッド距離）．** 長さ $l$ の 2 つのサブシーケンス $s_p$ および $s_q$ が与えられたとき，これらの $z$ 正規化ユークリッド距離 $d(s_p, s_q)$ は式 (2) により定義される．

$$d(s_p, s_q) = \sqrt{\sum_{i=1}^{l} \left( \frac{s_p[i] - \mu(s_p)}{\sigma(s_p)} - \frac{s_q[i] - \mu(s_q)}{\sigma(s_q)} \right)^2} \tag{2}$$

ここで，$\mu(s)$ および $\sigma(s)$ はそれぞれ $(s[1], s[2], ..., s[l])$ の平均および標準偏差である．$z$ 正規化ユークリッド距離とピアソン相関には以下の関係が成り立つ [11]．

$$\rho(s_p, s_q) = 1 - \frac{d(s_p, s_q)^2}{2l} \tag{3}$$

次に，時系列データの定義を多次元に拡張する．

**定義 4（多次元ストリーミング時系列データ）．** $d$ 次元の多次元ストリーミング時系列データ $\boldsymbol{t}$ は，同じ時刻に取得された各次元のストリーミング時系列データの集合であり，以下のように表現する．

$$\boldsymbol{t} = \begin{pmatrix} t^{(1)} \\ \vdots \\ t^{(d)} \end{pmatrix} = \begin{pmatrix} t^{(1)}[1],\ t^{(1)}[2],\ \dots \\ \vdots \\ t^{(d)}[1],\ t^{(d)}[2],\ \dots \end{pmatrix} \tag{4}$$

**定義 5（多次元サブシーケンス）．** $\boldsymbol{t}$ および長さ $l$ が与えられたとき，$p$ 番目の値を始点とする $\boldsymbol{t}$ の $d$ 次元の多次元サブシーケンス $\boldsymbol{s_p}$ は式 (5) により定義される．

$$\boldsymbol{s_p} = \begin{pmatrix} s_p^{(1)} \\ \vdots \\ s_p^{(d)} \end{pmatrix} = \begin{pmatrix} s_p^{(1)}[1],\ s_p^{(1)}[2],\ \dots,\ s_p^{(1)}[l] \\ \vdots \\ s_p^{(d)}[1],\ s_p^{(d)}[2],\ \dots,\ s_p^{(d)}[l] \end{pmatrix} \tag{5}$$

次に，多次元サブシーケンス間の距離を測るため，多次元サブシーケンス間の距離を定義する．1 章で述べたように，全ての次元を考慮すると有益な解が得られない場合がある．そのため，$d$ 次元の中から相関しない次元を除くため，$k$ 次元におけるサブシーケンス間の距離を定義する．

**定義 6（$k$ 次元におけるサブシーケンス間の距離）．** 長さ $l$ の 2 つの $d$ 次元サブシーケンス $\boldsymbol{s_p}$，$\boldsymbol{s_q}$，および $k\ (\leq d)$ が与えられたとき，$k$ 次元におけるこれらのサブシーケンス間の距離 $d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q})$ は式 (6) により定義される．

$$d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) = \min_i^{(k)} d(s_p^{(i)}, s_q^{(i)}) \tag{6}$$

ここで，$\min^{(k)}$ は $k$ 番目に小さい値を出力する関数とする．式 (3) と同様に，$k$ 次元におけるサブシーケンス間の

距離をピアソン相関に変換したものを定義する．

**定義 7（$k$ 次元におけるサブシーケンス間のピアソン相関）．** 長さ $l$ の 2 つの $d$ 次元サブシーケンス $\boldsymbol{s_p}$ および $\boldsymbol{s_q}$ の $k$ 次元における距離が $d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q})$ であるとき，$k$ 次元におけるこれらのサブシーケンスのピアソン相関 $\rho^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q})$ は式 (7) により定義される．

$$\rho^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) = 1 - \frac{d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q})^2}{2l} \tag{7}$$

次に，サブシーケンス $\boldsymbol{s_p}$ と類似する多次元サブシーケンス（類似サブシーケンス）を定義する．

**定義 8（類似サブシーケンス）．** $\boldsymbol{s_p}$，$\boldsymbol{s_q}$，$k$，およびある閾値 $\theta$ が与えられたとき，$\boldsymbol{s_p}(\boldsymbol{s_q})$ が $\boldsymbol{s_q}(\boldsymbol{s_p})$ と類似しているならば，次の条件を満たす．

$$\rho^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \geq \theta \Leftrightarrow d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \leq \sqrt{2l(1-\theta)} \tag{8}$$

$\boldsymbol{s_p}$ と $\boldsymbol{s_{p+1}}$ が互いに類似していることは自明であり，有用な結果を得るためには，このようなサブシーケンスを考慮すべきではない．そこで，互いに重なり合う多次元サブシーケンスをトリビアルマッチと定義する [7], [15]．

**定義 9（トリビアルマッチ）．** $\boldsymbol{s_p}$ が与えられたとき，$\boldsymbol{s_p}$ とトリビアルマッチである多次元サブシーケンスの集合 $S_p$ は次の条件を満たす．

$$S_p = \{\boldsymbol{s_q} \,|\, p - l + 1 \leq q \leq p + l - 1\} \tag{9}$$

ここで，時刻 $|\boldsymbol{t}|$ における全ての多次元サブシーケンスの数は $|\boldsymbol{t}| - l + 1$ であり，それらの集合を $S$ とする．このとき，多次元サブシーケンスの集合 $S$ に対して，ある多次元サブシーケンス $\boldsymbol{s_p}$ と類似した多次元サブシーケンスの数をスコアと定義する．

**定義 10（スコア）．** $\boldsymbol{t}$，$l$，$\theta$，および $k$ が与えられたとき，あるサブシーケンス $\boldsymbol{s_p}$ のスコアは式 (10) により定義される．

$$score(\boldsymbol{s_p}) = |\{\boldsymbol{s_q} \,|\, \boldsymbol{s_q} \in S \backslash S_p, \rho^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \geq \theta\}| \tag{10}$$

本研究では，このような環境においてスコアが最大となる多次元サブシーケンスをモニタリングする．つまり，本研究の問題は以下のように定義される．

**問題定義．** $\boldsymbol{t}$，$l$，$\theta$，および $k$ が与えられたとき，式 (11) で表される多次元レンジモチーフ $\boldsymbol{s^*}$ をモニタリングする．

$$\boldsymbol{s^*} = \underset{\boldsymbol{s_p} \in S}{\arg\max}\ score(\boldsymbol{s_p}) \tag{11}$$

### 2.2 ベースラインアルゴリズム

本稿は本問題に初めて取り組むため，まず，ベースラインとなるアルゴリズムについて考える．新たな値が観測された際，新たに生成される多次元サブシーケンスに対して，これまでに生成された全ての多次元サブシーケンスとの距離を計算し，スコアを更新する．そして，スコアが最大の多次元サブシーケンスをモチーフとする．前述したように，$\boldsymbol{t}$ には $|\boldsymbol{t}| - l + 1$ 個の多次元サブシーケンスが存在し，$k$ 次元におけるサブシーケンス間の距離の計算には $\mathcal{O}(dl)$ 時間かかる．そのため，ベースラインアルゴリズムの時間計算量は $\mathcal{O}((|\boldsymbol{t}| - l)dl)$ である．

ここで，ある多次元サブシーケンスのスコアに影響を与えるのは，その多次元サブシーケンスと類似した多次元サブシーケンスのみであるため，全ての多次元サブシーケンスのスコアを更新する必要はない．そのため，新たな値を取得したとき，スコアを更新する必要がある多次元サブシーケンスを効率的に特定するアルゴリズムを提案する．

## 3. 関連研究

時系列データマイニングに関する研究は多く行われている [1], [8], [10]．本章では，本研究に最も関連しているレンジモチーフおよび多次元モチーフに関する既存研究についてのみ紹介する．

### 3.1 レンジモチーフ

文献 [15] では，レンジモチーフを効率的に発見するための近似アルゴリズムを提案している．このアルゴリズムでは，各サブシーケンスを SAX（Symbolic Aggregate approXimation）を用いて記号列に変換する．このアルゴリズムと同様に，文献 [5] では，$i$SAX（indexable SAX）を用いてレンジモチーフを発見するアルゴリズムを提案している．SAX および $i$SAX は時系列データを記号列に近似するため，発見されたモチーフが正確であることが保証されない．また，いくつかの確率的アルゴリズムが提案されているが [7], [18]，これらのアルゴリズムも発見されたモチーフが正確であることは保証されない．文献 [9] では，スライディングウィンドウ上でレンジモチーフをモニタリングする問題に取り組んでいる．文献 [9] の提案アルゴリズムでは，ウィンドウ内のサブシーケンスを Piecewize Aggregate Approximation で圧縮後，kd 木で管理している．kd 木を用いた範囲検索によりレンジモチーフが更新されるかどうかを高速に把握できる．これらの研究は 1 次元の時系列データを対象としており，多次元時系列データは対象としていない．

### 3.2 多次元モチーフ

文献 [16] では，多次元時系列データを Principal Component Analysis で 1 次元時系列データに変換し，モチーフの発見を行う．モチーフ発見の精度および速度は，5 つのパラメータの調整が必要であり，実践的でない．文献 [13] では，無関係な次元を除いた次元に対してモチーフ発見を行

うことで，ノイズに強く，また，有益なモチーフを発見できるとしている．具体的には，各次元のサブシーケンスをSAX を用いて記号列に変換する．また，各次元間に対して，記号列に変換されたサブシーケンス間の距離がある閾値を超えるまで有効な次元であるとして，無関係な次元を除外する．文献 [19] では，Matrix Profile と呼ばれるデータ構造を用いて多次元モチーフを発見するアルゴリズム $m$STAMP を提案している．このデータ構造は，全てのサブシーケンスに対して最近傍のサブシーケンスとの距離を保持する．$m$STAMP は，全次元の組合せに対してモチーフ発見を行い，Minimum Description Length を用いて，モチーフ発見に用いる最適な次元を決定している．これらの研究は静的な多次元時系列データを対象としている．

## 4. MMM: Multidimensional Motif Monitoring

新たな値が観測された際，これまでに生成された多次元サブシーケンスのスコアは最大で 1 増加する．そのため，モチーフは頻繁に変化せず，新たに生成される多次元サブシーケンスのスコアが頻繁に $score(\boldsymbol{s^*})$ を超えることは非常にまれである．

ここで，新たに生成されるサブシーケンスを $\boldsymbol{s_n}$ としたとき，高速に $score(\boldsymbol{s_n}) < score(\boldsymbol{s^*})$ であることがわかれば，正確なモチーフを効率的にモニタリングできる．これを実現するため，4.1 節においてある次元における距離計算の回数を削減するアルゴリズムを提案し，4.2 節において $score(\boldsymbol{s_n})$ の上界値を取得するアルゴリズムを提案する．最後に，4.3 節において MMM の全体的なアルゴリズムを紹介し，MMM の計算量について述べる．

### 4.1 距離計算回数の削減

まず，これ以降に必要となる重要な定理を紹介する．

**定理 1（類似サブシーケンス）.** 以下の条件を満たすとき，$\boldsymbol{s_p}$ と $\boldsymbol{s_q}$ は類似サブシーケンスである．

$$|\{i \mid 1 \leq i \leq d, d(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}\}| \geq k \quad (12)$$

**証明.** $d(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}$ を満たす $i$ が $k$ 個以上あるとき，$k$ 番目に小さい $d(s_p^{(i)}, s_q^{(i)})$ は必ず $\sqrt{2l(1-\theta)}$ 以下である．したがって，$d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \leq \sqrt{2l(1-\theta)}$ が成り立つため，$\boldsymbol{s_p}$ と $\boldsymbol{s_q}$ は類似サブシーケンスである． □

つまり，各次元 $i$ のサブシーケンス $s_p^{(i)}$ および $s_q^{(i)}$ に対して，$d(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}$ を満たす次元数を計算し，その数が $k$ 以上かどうかを調べることで，$\boldsymbol{s_p}$ と $\boldsymbol{s_q}$ が類似サブシーケンスであるかどうかを把握できる．そのため，ある次元 $i$ におけるサブシーケンス間の距離計算回数を削減する．

長さ $l$ のサブシーケンス $s_p^{(i)}$ は $l$ 次元上の点として表現できる．このとき，定理 2 が成り立つ．

**定理 2（$z$ 正規化ユークリッド距離の下界値・上界値）.** ある次元 $i$ の 3 つのサブシーケンス $s_p^{(i)}$，$s_q^{(i)}$，および $s_r^{(i)}$ に対して，以下の不等式が成り立つ．

$$\begin{aligned} |d(s_p^{(i)}, s_r^{(i)}) - d(s_q^{(i)}, s_r^{(i)})| &\leq d(s_p^{(i)}, s_q^{(i)}) \\ &\leq d(s_p^{(i)}, s_r^{(i)}) + d(s_q^{(i)}, s_r^{(i)}) \end{aligned} \quad (13)$$

**証明.** 三角不等式より定理 2 が成り立つ． □

$d(s_p^{(i)}, s_r^{(i)})$ および $d(s_q^{(i)}, s_r^{(i)})$ が事前に分かっているとき，定理 2 より，$d(s_p^{(i)}, s_q^{(i)})$ の下界値 $d_{lb}(s_p^{(i)}, s_q^{(i)})$ および上界値 $d_{ub}(s_p^{(i)}, s_q^{(i)})$ を $\mathcal{O}(1)$ で得ることができる．$d_{lb}(s_p^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ である場合，$d(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}$ を満たさないため，正確性を失うことなく $s_p^{(i)}$ と $s_p^{(i)}$ の正確な距離計算を枝刈りできる．また，$d_{ub}(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}$ を満たす場合，正確な距離計算をすることなく，$d(s_p^{(i)}, s_q^{(i)}) \leq \sqrt{2l(1-\theta)}$ を満たすことがわかる．

ここで，あるサブシーケンスを中心とするサブシーケンスの集合（クラスタ）を定義する．

**定義 11（クラスタ）.** クラスタ $C_p^{(i)}$ は，サブシーケンス $s_p^{(i)}$ を中心とし，$s_p^{(i)}$ との距離が $r$ 以下であるサブシーケンス $s_q^{(i)}$ の集合であり，$s_q^{(i)}$ は $s_p^{(i)}$ との距離の降順にソートされている．

新たに生成されたサブシーケンス $s_n^{(i)}$，クラスタ $C_p^{(i)}$ の中心サブシーケンス $s_p^{(i)}$，および $C_p^{(i)}$ 内のサブシーケンス $s_q^{(i)}$ に対して定理 2 が成り立つ．このとき，定理 3 が成り立つ．

**定理 3（距離計算の打ち切り）.** 新たに生成されたサブシーケンス $s_n^{(i)}$，距離の閾値 $\sqrt{2l(1-\theta)}$，および $s_p^{(i)}$ を中心とするクラスタ $C_p^{(i)}$ が与えられたとし，$d(s_n^{(i)}, s_p^{(i)})$ は事前に分かっているとする．$d(s_n^{(i)}, s_p^{(i)})$ が $C_p^{(i)}$ のクラスタ半径（中心から最も遠いサブシーケンスとの距離）よりも大きいならば，$s_q^{(i)} \in C_p^{(i)}$ に対して $d_{lb}(s_n^{(i)}, s_q^{(i)})$ を順に計算したとき，$d_{lb}(s_n^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ を満たした時点で，それ以降の距離計算を打ち切っても正確性は失われない．

**証明.** クラスタ内のサブシーケンス $s_q^{(i)} \in C_p^{(i)}$ は中心 $s_p^{(i)}$ との距離の降順にソートされているため，定理 2 により計算される $d_{lb}(s_n^{(i)}, s_q^{(i)})$ は単調に増加する．そのため，$d_{lb}(s_n^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ を満たした場合，それ以降も常に $d_{lb}(s_n^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ が成り立つ．$d_{lb}(s_n^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ が成り立つとき，$d(s_p^{(i)}, s_q^{(i)}) > \sqrt{2l(1-\theta)}$ である．以上より，定理 3 が成り立つ． □

**例 1.** 図 2 は，長さ 2 のサブシーケンスを 2 次元上の点として表現しており，距離の閾値を 5 とする．また，$s_p^{(i)}$ と $s_p^{(i)}$ を中心とするクラスタ $C_p^{(i)}$ 内のサブシーケンスとの距離を表に示す．$d(s_n^{(i)}, s_p^{(i)}) = 10$ であると

| $C_p^{(i)}$ | $d(s_p^{(i)}, s_\cdot^{(i)})$ | $d_{lb}(s_p^{(i)}, s_\cdot^{(i)})$ |
|---|---|---|
| $s_e^{(i)}$ | 6.2 | 10 − 6.2 = 3.8 |
| $s_b^{(i)}$ | 4.5 | 10 − 4.5 = 5.5 |
| $s_a^{(i)}$ | 3.8 | × |
| $s_c^{(i)}$ | 3.7 | × |
| $s_d^{(i)}$ | 2.9 | × |

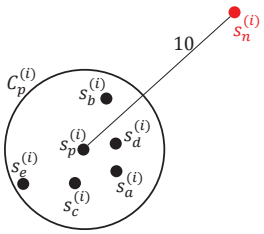



図 2: 距離計算の打ち切り

き，$d_{lb}(s_n^{(i)}, s_e^{(i)}) = 10 - 6.2 = 3.8 < 5$ である．次に，$d_{lb}(s_n^{(i)}, s_b^{(i)}) = 10 - 4.5 = 5.5 > 5$ であるため，これ以降の計算を打ち切ることができる．

新たに生成されたサブシーケンス $s_n^{(i)}$ と全クラスタに対して定理 3 を用いることで，$s_n^{(i)}$ とこれまでに生成されたサブシーケンスとの距離計算回数を削減できる．

### 4.2 スコアの上界値の取得

定理 1 より，以下の系が成り立つ．

**系 1.** $|\{i \,|\, 1 \le i \le d, d_{lb}(s_p^{(i)}, s_q^{(i)}) \le \sqrt{2l(1-\theta)}\}| \ge k$ が成り立つとき，$\boldsymbol{s_p}$ と $\boldsymbol{s_q}$ は類似する可能性がある．

新たに生成された多次元サブシーケンス $\boldsymbol{s_n}$ とこれまでに生成された全ての多次元サブシーケンス $\boldsymbol{s_p}$ に対して，系 1 を満たすサブシーケンスの数が $score(\boldsymbol{s_n})$ の上界値となる．$\boldsymbol{s_n}$ のスコアの上界値が $score(\boldsymbol{s}^*)$ を超えた場合，$\boldsymbol{s_n}$ の正確なスコアを計算しモチーフが更新されるかどうかを確認する必要がある．このとき，$\boldsymbol{s_n}$ と類似する可能性のあるサブシーケンスとのみ距離計算することで効率的にモチーフを更新できる．そこで，あるサブシーケンス $\boldsymbol{s_p}$ と類似する可能性のあるサブシーケンスを保存するリスト PL（**P**ossible Similar Subsequence **L**ist）を定義する．

**定義 12（PL）.** $\boldsymbol{s_p}$ の PL $PL_p$ は，サブシーケンスの識別子 $q$ の集合であり，$PL_p$ に含まれるあるサブシーケンス $\boldsymbol{s_q}$ は以下の条件を満たす．

$$\boldsymbol{s_q} \in S \backslash S_p,\ d_{lb}^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \le \sqrt{2l(1-\theta)} \qquad (14)$$

ここで，$d_{lb}^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q})$ は $k$ 次元における $\boldsymbol{s_p}$，$\boldsymbol{s_q}$ 間の距離の下界値とする．さらに，$PL_p$ に含まれるサブシーケンスと距離計算を行い，$d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \le \sqrt{2l(1-\theta)}$ となった $\boldsymbol{s_q}$ の数を暫定のスコアとする．

**定義 13（暫定のスコア）.** $\boldsymbol{s_p}$ の暫定のスコア $score_{tmp}(\boldsymbol{s_p})$ は以下の条件を満たすサブシーケンスの数である．

$$\boldsymbol{s_q} \in S \backslash S_p,\ d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \le \sqrt{2l(1-\theta)},\ q \notin PL_p \quad (15)$$

つまり，$\boldsymbol{s_p}$ の正確なスコアを計算するとき，$PL_p$ から $q$ を取り出し，$d^{(k)}(\boldsymbol{s_p}, \boldsymbol{s_q}) \le \sqrt{2l(1-\theta)}$ ならば $score_{tmp}(\boldsymbol{s_p})$ を 1 増やす．そして，$score_{tmp}(\boldsymbol{s_p}) + |PL_p| < score(\boldsymbol{s}^*)$ となった時点で，$\boldsymbol{s_p}$ はモチーフにならないため，距離計算を終了する．

### 4.3 アルゴリズム

本節では MMM の詳細を紹介する．ある時刻 $|\boldsymbol{t}|$ において前処理を行ってから，MMM を実行する．

**前処理.** k-means++[2] のクラスタの中心の決定方法に基づき，事前にクラスタを作成する．ある時刻 $|\boldsymbol{t}|$ において，各次元 $i$ に対して以下の処理を行う．まず，全てのサブシーケンスの平均の距離 $d_{avg}^{(i)}$ を計算する．次に，ランダムにサブシーケンスを選択し，そのサブシーケンスを中心とする半径 $r = d_{avg}^{(i)} - \sqrt{2l(1-\theta)}$ のクラスタを作成する．その後，以下の手順を全てのサブシーケンスがクラスタに含まれるまで繰り返す．全てのサブシーケンスに対して最近傍クラスタとの距離を計算し，まだクラスタリングされていないサブシーケンスの中から，重み付き確率分布によりサブシーケンスを選択する．まだクラスタリングされていないサブシーケンスに対して，選択されたサブシーケンスを中心とする半径 $r$ のクラスタを作成する．

**MMM.** まず，新たな多次元サブシーケンス $\boldsymbol{s_n}$ が生成されたとき，各次元 $i$ に対して処理を行う．$s_n^{(i)}$ がどのクラスタに含まれるかを決めるため，$s_n^{(i)}$ の最近傍のクラスタの識別子および中心との距離を保存する $cluster\_id$ および $cluster\_dist$ を初期化する（2 行）．次に，次元 $i$ に存在するクラスタの集合 $\mathcal{C}^{(i)}$ に含まれるクラスタ $C_j^{(i)}$ に対して処理を行う．$s_n^{(i)}$ とクラスタの中心のサブシーケンス $s_j^{(i)}$ との距離を計算する（4 行）．そして，$s_n^{(i)}$ の最近傍のクラスタを必要に応じて更新する（5–6 行）．$C_j^{(i)}$ 内のサブシーケンス $s_p^{(i)}$ に対して，$|dist - d(s_p^{(i)}, s_j^{(i)})|$ の昇順に処理を行う．ここで，$\boldsymbol{s_n}$ と $\boldsymbol{s_p}$ に対して類似する次元を一時的に保存する $rd_{n,p}$，および類似する可能性のある次元を一時的に保存する $prd_{n,p}$ を初期化する（8 行）．$s_n^{(i)}$ と $s_p^{(i)}$ の距離の下界値 $dist_{lb}$ を定理 2 を用いて計算する（9 行）．$dist_{lb} \le \sqrt{2l(1-\theta)}$ である場合，$s_n^{(i)}$ と $s_p^{(i)}$ の距離の上界値 $dist_{ub}$ を定理 2 を用いて計算し，さらに $dist_{ub} \le \sqrt{2l(1-\theta)}$ である場合，$rd_{n,p}$ に $i$ を追加し，そうでない場合，$prd_{n,p}$ に $i$ を追加する（10–15 行）．$dist_{lb} > \sqrt{2l(1-\theta)}$ である場合，定理 3 より，それ以降の計算を打ち切る（16–17 行）．$\mathcal{C}^{(i)}$ に含まれる $C_j^{(i)}$ に対しての処理を終えると，$s_n^{(i)}$ をクラスタに追加する処理を行う．$cluster\_dist \le d_{avg}^{(i)} - \sqrt{2l(1-\theta)}$ である場合，$C_{cluster\_id}^{(i)}$ に $s_n^{(i)}$ を追加する（18–19 行）．そうでない場合，$s_n^{(i)}$ を中心とするクラスタ $C_n^{(i)}$ を作成し，$\mathcal{C}^{(i)}$ に追加する（20–21 行）．

次に，$\boldsymbol{s_n}$ の暫定のスコア $score_{tmp}(\boldsymbol{s_n})$，および $\boldsymbol{s_n}$ の PL $PL_n$ を初期化する（22 行）．これまでに生成された全てのサブシーケンス $\boldsymbol{s_p}$ に対して，$\boldsymbol{s_n}$ と類似するかどうかを確

認する．$|rd_{n,p}|+|prd_{n,p}| \geq k$ を満たし，さらに $|rd_{n,p}| \geq k$ を満たす場合，$\boldsymbol{s_n}$ と $\boldsymbol{s_p}$ は類似するため，$score_{tmp}(\boldsymbol{s_n})$ および $score(\boldsymbol{s_p})$ を 1 増加させる（24–26 行）．$|rd_{n,p}| \geq k$ を満たさない場合，$\boldsymbol{s_n}$ と $\boldsymbol{s_p}$ は類似する可能性があるため，$PL_n$ および $PL_p$ にそれぞれ $p$ および $n$ を追加する（27–28 行）．$score_{tmp}(\boldsymbol{s_p})$ または $|PL_p|$ が増加し，スコアの上界値が $score(\boldsymbol{s}^*)$ 超えるとき，Motif-Update を実行する（29–30 行）．Motif-Update の詳細は後述する．

これまでに生成された全てのサブシーケンス $\boldsymbol{s_p}$ に対しての処理を終え，$score_{tmp}(\boldsymbol{s_n}) + |PL_n| > score(\boldsymbol{s}^*)$ である場合，Motif-Update を実行する（31–32 行）．最後に，$S$ に $\boldsymbol{s_n}$ を追加する（33 行）．

**Motif-Update.** Motif-Update$(\boldsymbol{s_p}, \boldsymbol{s}^*)$ はモチーフを更新するアルゴリズムである．$score_{tmp}(\boldsymbol{s_p})+|PL_p| > score(\boldsymbol{s}^*)$ 満たす場合，$\boldsymbol{s_p}$ はモチーフになり得るため，$\boldsymbol{s_p}$ の正確なスコアを計算する必要がある．そのため，$PL_p$ に含まれる多次元サブシーケンスとの $k$ 次元における距離を計算することで，$\boldsymbol{s_p}$ の正確なスコアを計算する．$\boldsymbol{s_p}$ のスコアが $score(\boldsymbol{s}^*)$ より大きい場合，$\boldsymbol{s_p}$ がモチーフとして返却され，それ以外の場合，$\boldsymbol{s}^*$ が返却される．

**時間計算量.** アルゴリズム 1 における 21 行目までの処理には，平均クラスタ数を $c$，距離計算を打ち切るまでの平均繰り返し回数を $c'$ としたとき，$\mathcal{O}(dc(l+c'))$ 時間かかる．ここで，$\boldsymbol{s_p}$ のスコアを正確に計算する場合，$|PL_p|$ 回距離計算が必要となるため $\mathcal{O}(|PL_p|dl)$ 時間かかる．よって，新たな値を取得した際，正確なスコア計算が必要なサブシーケンスの集合を $S'$ としたとき，アルゴリズム 1 における 22 行目以降の処理には，$\mathcal{O}(\sum_{S'}|PL_p|dl)$ 時間かかる．よって，MMM の時間計算量は $\mathcal{O}(dc(l+c')+\sum_{S'}|PL_p|dl)$ となる．

**Algorithm 1:** MMM

**Input:** $\boldsymbol{s_n}$: the new subsequence
**Output:** $\boldsymbol{s}^*$: the discord

1 **for** $i = 1$ *to* $d$ **do**
2  $cluster\_id \leftarrow -1,\ cluster\_dist \leftarrow \infty$
3  **for** $\forall C_j^{(i)} \in \mathcal{C}^{(i)}$ **do**
4  $dist \leftarrow d(s_n^{(i)}, s_j^{(i)})$
5  **if** $dist < cluster\_dist$ **then**
6  $cluster\_id \leftarrow j,\ cluster\_dist \leftarrow dist$
7  **for** $\forall s_p^{(i)} \in C_j^{(i)} \backslash S_n$ *such that* $|dist - d(s_p^{(i)}, s_j^{(i)})|$ *is acsending order* **do**
8  $rd_{n,p} \leftarrow \varnothing,\ prd_{n,p} \leftarrow \varnothing$
9  $dist_{lb} \leftarrow |dist - d(s_p^{(i)}, s_j^{(i)})|$
10  **if** $dist_{lb} \leq \sqrt{2l(1-\theta)}$ **then**
11  $dist_{ub} \leftarrow dist + d(s_p^{(i)}, s_j^{(i)})$
12  **if** $dist_{ub} \leq \sqrt{2l(1-\theta)}$ **then**
13  $rd_{n,p} \leftarrow rd_{n,p} \cup \{i\}$
14  **else**
15  $prd_{n,p} \leftarrow prd_{n,p} \cup \{i\}$
16  **else**
17  break
18  **if** $cluster\_dist \leq d_{avg}^{(i)} - \sqrt{2l(1-\theta)}$ **then**
19  $C_{cluster\_id}^{(i)} \leftarrow C_{cluster\_id}^{(i)} \cup \{s_n^{(i)}\}$
20  **else**
21  $C_n^{(i)} \leftarrow \{s_n^{(i)}\},\ \mathcal{C}^{(i)} \leftarrow \mathcal{C}^{(i)} \cup \{C_n^{(i)}\}$
22 $score_{tmp}(\boldsymbol{s_n}) \leftarrow 0,\ PL_n \leftarrow \varnothing$
23 **for** $\forall \boldsymbol{s_p} \in S$ **do**
24  **if** $|rd_{n,p}| + |prd_{n,p}| \geq k$ **then**
25  **if** $|rd_{n,p}| \geq k$ **then**
26  $score_{tmp}(\boldsymbol{s_n}) \leftarrow score_{tmp}(\boldsymbol{s_n}) + 1,$ $score_{tmp}(\boldsymbol{s_p}) \leftarrow score_{tmp}(\boldsymbol{s_p}) + 1$
27  **else**
28  $PL_n \leftarrow PL_p \cup \{p\},\ PL_p \leftarrow PL_p \cup \{n\}$
29  **if** $score_{tmp}(\boldsymbol{s_n}) + |PL_p| > score(\boldsymbol{s}^*)$ **then**
30  $\boldsymbol{s}^* \leftarrow$Motif-Update$(\boldsymbol{s_p}, \boldsymbol{s}^*)$
31 **if** $score_{tmp}(\boldsymbol{s_n}) + |PL_n| > score(\boldsymbol{s}^*)$ **then**
32  $\boldsymbol{s}^* \leftarrow$Motif-Update$(\boldsymbol{s_n}, \boldsymbol{s}^*)$
33 $S \leftarrow S \cup \{\boldsymbol{s_n}\}$

## 5. 評価実験

本章では，MMM およびベースラインアルゴリズムの性能評価のために行った実験の結果を紹介する．

### 5.1 実験環境

全ての実験は，Windows 10 Pro，3.00GHz Intel Xeon Gold，および 512GB RAM を搭載した計算機で行い，全てのアルゴリズムを C++で実装した．

**データセット.** 以下の 4 つの実データを用いた．

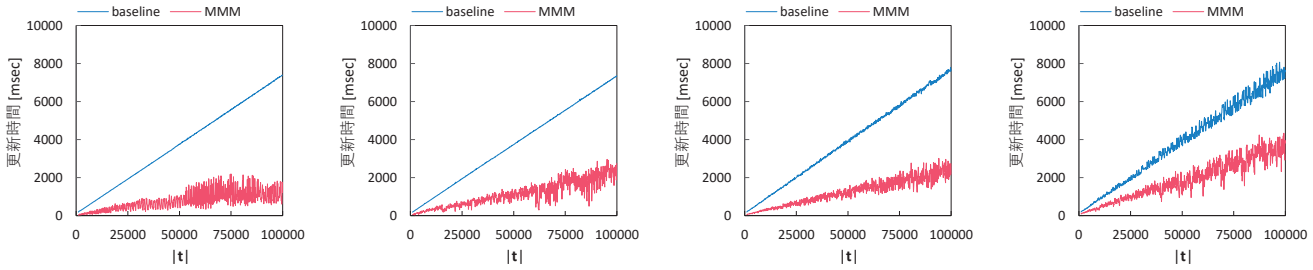
- Cricket [6]：加速度センサの多次元時系列データ（6 次元）
- EigenWorms [6]：線虫の動きの多次元時系列データ（6 次元）
- SelfRegulationSCP1 [6]：脳波の多次元時系列データ（6 次元）
- HousePowerConsumption[*1]：フランスのある家庭における消費電力の多次元時系列データ（7 次元）

**パラメータ.** 本実験で用いたパラメータを表 1 に示す．太字で表されている値はデフォルトの値であり，あるパラメータの影響を調べるとき，他のパラメータは固定する．

**評価指標.** モチーフの更新時間，および時刻 100,000 までにかかったモチーフの更新時間の合計を評価する．

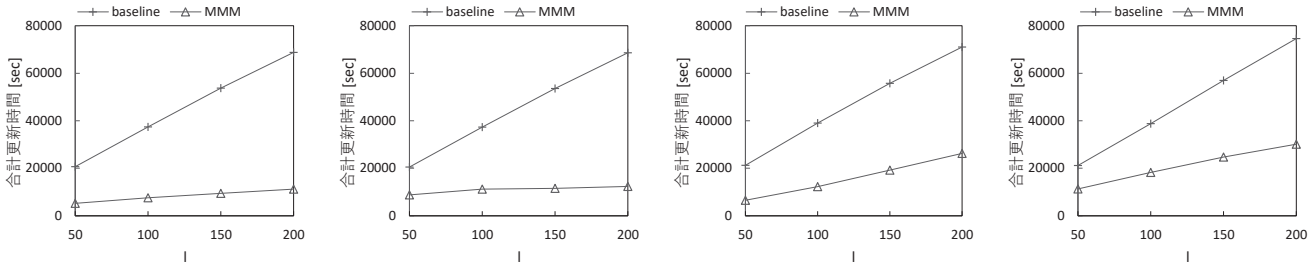
**初期状態.** $|\boldsymbol{t}| = 1{,}000$ において前処理を行った状態で実験

[*1] http://archive.ics.uci.edu/ml/datasets.php




(a) Cricket (b) EigenWorms (c) SelfRegulationSCP1 (d) HousePowerConsumption

図 3: $|\boldsymbol{t}|$ の影響




(a) Cricket (b) EigenWorms (c) SelfRegulationSCP1 (d) HousePowerConsumption

図 4: $l$ の影響

表 1: パラメータ設定

| パラメータ | 値 |
|---|---|
| $\|\boldsymbol{t}\|$ | 1,000~100,000 |
| $l$ | 50, **100**, 150, 200 |
| $\theta$ | 0.75, 0.8, 0.85, **0.9**, 0.95 |
| $k$ | 2, **3**, 4 |

を開始する．

### 5.2 評価結果

**$|\boldsymbol{t}|$ の影響．** 図 3 に $|\boldsymbol{t}|$ を変化させたときの結果を示す．MMM の更新時間は，ベースラインアルゴリズムよりも高速である．これは，ベースラインアルゴリズムは新たに生成された多次元サブシーケンスとこれまでに生成された全ての多次元サブシーケンスとの正確な距離計算を行っているが，MMM はモチーフが更新される可能性があるときのみ正確な距離計算を行っているからである．どちらのアルゴリズムにおける更新時間も，$|\boldsymbol{t}|$ の増加に伴い，線形に増加する．これは，$|\boldsymbol{t}|$ の増加によって，距離計算を行う多次元サブシーケンスの数が増加するからである．

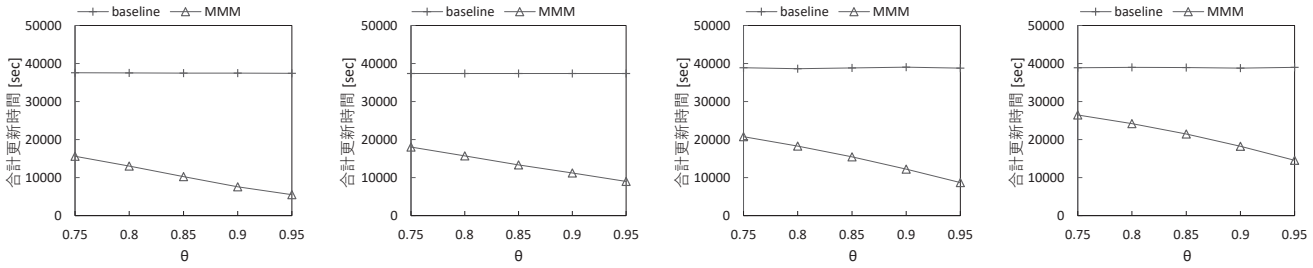
**$l$ の影響．** 図 4 に $l$ を変化させたときの結果を示す．$|\boldsymbol{t}|$ を変化させたときと同様の理由でベースラインアルゴリズムよりも MMM が高速となっている．どちらのアルゴリズムにおいても，$l$ の増加に伴って合計更新時間は増加する．これは，$l$ の増加によって，距離計算にかかる時間が増加するからである．

**$\theta$ の影響．** 図 5 に $\theta$ を変化させたときの結果を示す．ベースラインアルゴリズムでは，新たな値を取得した際，これまでに生成された全ての多次元サブシーケンスとの距離を計算するため，合計更新時間は $\theta$ によらず一定である．これは，ベースラインアルゴリズムの時間計算量が $\mathcal{O}((|\boldsymbol{t}|-l)dl)$ であることから明らかである．一方で，MMM は $\theta$ の増加に伴って合計更新時間は減少する．これは，$\theta$ の増加に伴って，距離の閾値が小さくなり，早い段階で定理 3 による距離計算の打ち切りが行われるからである．また，$\theta$ の増加に伴って，類似する可能性のある多次元サブシーケンスの数が減少し，Motif-Update 実行時の正確な距離計算の回数が減少する．

**$k$ の影響．** 図 6 に $k$ を変化させたときの結果を示す．図 5 と同様の結果が得られていることがわかる．ベースラインアルゴリズムでは，$\theta$ を変化させたときと同様の理由で合計更新時間は $k$ によらず一定である．一方，MMM は $k$ の増加に伴って合計更新時間は減少する．これは，$k$ の増加に伴って，$k$ 次元におけるサブシーケンス間の距離が大きくなり，早い段階で定理 3 による距離計算の打ち切りが行われるからである．また，$k$ の増加に伴って，類似する可能性のある多次元サブシーケンスの数が減少し，Motif-Update 実行時の正確な距離計算の回数が減少する．

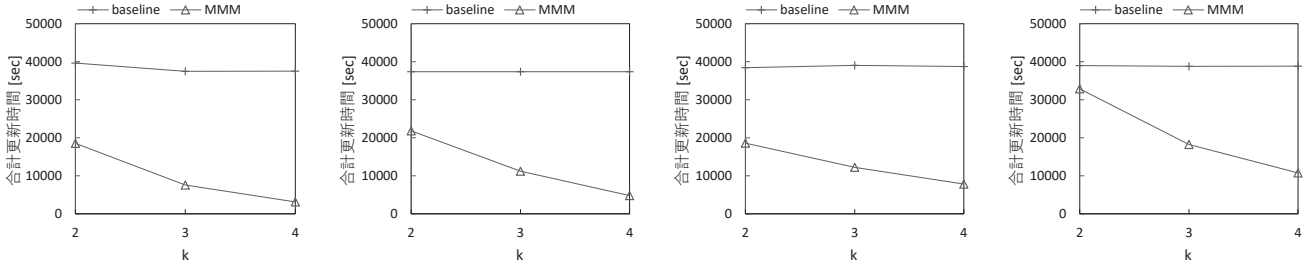
## 6. 結論

近年，多くの多次元ストリーミング時系列データが生成されており，それらをリアルタイムに解析することが重要になっている．本稿では，多次元ストリーミング時系列データに対して多次元レンジモチーフをモニタリングする問題に初めて取り組んだ．効率的にモチーフをモニタリングするため，MMM を提案した．MMM は各次元のサブ




(a) Cricket (b) EigenWorms (c) SelfRegulationSCP1 (d) HousePowerConsumption

図 5: $\theta$ の影響




(a) Cricket (b) EigenWorms (c) SelfRegulationSCP1 (d) HousePowerConsumption

図 6: $k$ の影響

シーケンスをクラスタに分割し，三角不等式を用いることで不要な距離計算を削減することができる．実データを用いた評価実験により，MMM の有効性を確認した．

**謝辞** 本研究の一部は，基盤研究 (A)(18H04095)，基盤研究 (B)(JP17KT0082)，および若手研究 (B)(JP16K16056) の研究助成によるものである．ここに記して謝意を表す．

## 参考文献

[1] Amagata, D. and Hara, T.: Mining top-k co-occurrence patterns across multiple streams, *TKDE*, Vol. 29, No. 10, pp. 2249–2262 (2017).

[2] Arthur, D. and Vassilvitskii, S.: k-means++: The advantages of careful seeding, *SODA*, pp. 1027–1035 (2007).

[3] Balasubramanian, A., Wang, J. and Prabhakaran, B.: Discovering multidimensional motifs in physiological signals for personalized healthcare, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 10, No. 5, pp. 832–841 (2016).

[4] Berlin, E. and Van Laerhoven, K.: Detecting leisure activities with dense motif discovery, *UbiComp*, pp. 250–259 (2012).

[5] Castro, N. and Azevedo, P.: Multiresolution motif discovery in time series, *SDM*, pp. 665–676 (2010).

[6] Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A. and Batista, G.: The UCR Time Series Classification Archive (2015). `www.cs.ucr.edu/~eamonn/time_series_data/`.

[7] Chiu, B., Keogh, E. and Lonardi, S.: Probabilistic discovery of time series motifs, *KDD*, pp. 493–498 (2003).

[8] Esling, P. and Agon, C.: Time-series data mining, *ACM Computing Surveys*, Vol. 45, No. 1, p. 12 (2012).

[9] Kato, S., Amagata, D., Nishio, S. and Hara, T.: Monitoring Range Motif on Streaming Time-Series, *DEXA*, pp. 251–266 (2018).

[10] Kato, S., Amagata, D., Nishio, S. and Hara, T.: Discord Monitoring for Streaming Time-series, *DEXA*, pp. 79–94 (2019).

[11] Li, Y., Yiu, M. L., Gong, Z. et al.: Discovering longest-lasting correlation in sequence databases, *PVLDB*, Vol. 6, No. 14, pp. 1666–1677 (2013).

[12] Linardi, M., Zhu, Y., Palpanas, T. and Keogh, E.: Matrix profile X: Valmod-scalable discovery of variable-length motifs in data series, *SIGMOD*, pp. 1053–1066 (2018).

[13] Minnen, D., Isbell, C., Essa, I. and Starner, T.: Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery, *ICDM*, pp. 601–606 (2007).

[14] Mueen, A., Keogh, E., Zhu, Q., Cash, S. and Westover, B.: Exact discovery of time series motifs, *SDM*, pp. 473–484 (2009).

[15] Patel, P., Keogh, E., Lin, J. and Lonardi, S.: Mining motifs in massive time series databases, *ICDM*, pp. 370–377 (2002).

[16] Tanaka, Y., Iwamoto, K. and Uehara, K.: Discovery of time-series motif from multi-dimensional data based on MDL principle, *Machine Learning*, Vol. 58, No. 2-3, pp. 269–300 (2005).

[17] Vahdatpour, A., Amini, N. and Sarrafzadeh, M.: Toward unsupervised activity discovery using multi dimensional motif detection in time series, *IJCAI*, pp. 1261–1266 (2009).

[18] Yankov, D., Keogh, E., Medina, J., Chiu, B. and Zordan, V.: Detecting time series motifs under uniform scaling, *KDD*, pp. 844–853 (2007).

[19] Yeh, C.-C. M., Kavantzas, N. and Keogh, E.: Matrix profile VI: Meaningful multidimensional motif discovery, *ICDM*, pp. 565–574 (2017).

[20] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A. and Keogh, E.: Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets, *ICDM*, pp. 1317–1322 (2016).