3K4-4in1

行動データマイニングのためのオンライン離散化手法の提案

Online Discretization and Pattern Mining in Time-Series Human Activity Data Stream

吉田一生 *1　山本泰生 *2,*3　岩沼宏治 *2
Kazuki Yoshida　Yoshitaka Yamamoto　Koji Iwanuma

*1山梨大学工学部コンピュータ理工学科
Department of Computer Science, Faculty of Engineering, University of Yamanashi

*2山梨大学大学院医学工学総合研究部
Interdisciplinary Graduate School of Medical and Engineering, University of Yamanashi

*3科学技術振興機構 さきがけ
Strategic Basic Research Programs PRESTO

In this paper, we present a novel technique based on online pattern mining for identifying human activities from motion sensing data. Our pattern mining technique enables to discover *motifs* of unknown human activities classes. However, discretization is essential for pattern mining. It is also useful for data compression to embed online discretization into our methodology. As a result, the compression rate of streaming data is 99.8% by our technique. The success rate of activity identirication is 77% on average compared the previous technique.

1. はじめに

スマートフォンのようなセンサを内蔵した小型計算機の出現により，人間行動に関する時系列情報を含んだ大規模で多様なストリームデータを収集することが可能になった．近年では，多様な行動を機械により識別する研究が盛んにおこなわれており，人間行動の理解に基づいたサービスの実現に役立てられている．例えば大内ら [6] はスマートフォンの GPS 機能・音センサ・加速度センサを用いて行動認識を実現しており，高齢者向けの見守りサービスといった応用が期待されている．

行動クラスは多岐にわたるため，従来の手法では人手での特徴量の設計が困難といった問題があった．そこで先行研究 [1] では，加速度データに頻出する時系列パターンを素性として用いることで行動識別を実現する手法が提案された．時系列情報を含んだ素性の自動設計を行うことで, 前述した従来の手法における問題点の解決を図っている．

本研究では，先行研究 [1] で実現できていなかった，加速度情報の離散化処理部のオンライン化を目的とする．先行研究では，数値データである入力データの前加工として離散化処理を行っている．この際，入力された全てのデータを記憶することで離散化時の境界点を算出しており，オンライン化には対応していなかった．そこで本研究ではストリームデータに対し，Zhang ら [4] により提案されたオンライン分位数計算アルゴリズムを適用する．入力データを許容誤差の範囲内で圧縮し，分位数を離散化時の境界点として求めることで，オンラインでの離散化処理を実現する．

2. 先行研究

本章では, 先行研究 [1] により提案された，オンラインマイニングによる行動識別アルゴリズムについて述べる．行動識別手法の概要を述べた後，各要素技術について述べる．

連絡先: 山梨大学工学部コンピュータ理工学科，〒 400-8510 山梨県甲府市武田 4-4-37 山梨大学甲府キャンパス A3 号館北 2 階 K221，t12cs062@yamanashi.ac.jp

1. 入力加速度データをすべて読み込み，離散化のための境界点を定め，再度走査を行い加速度データを離散化する
2. 離散化済み時系列データに対し，x・y・z 各軸に対して素性となる頻出系列パターン集合を抽出する
3. 抽出された頻出系列パターン集合と各行動クラス毎の頻出系列パターン集合データとの類似度を計算する
4. 算出されたデータの類似度から，k 近傍法により行動クラスを同定する

2.1 離散化処理

加速度センサ値は連続値であり，カテゴリ属性を対象とする一般的なパターンマイニング手法で扱うには適していない．そこで，加速度センサ値を離散化することで，パターンマイニング手法を適用できるようデータ加工を行う．

ここで離散化処理には，1 つの区間幅が等しくなる等幅分割法，1 つの区間に含まれるデータ量が等しくなる等深分割法が考えられ，今回は等深分割法による離散化処理を対象とする．

はじめに対象となるセンサデータをすべて走査し，データの分布を把握することで，離散化のための境界点を定める．境界点が定まったら，再度センサデータをすべて走査し，定めた境界点との比較を行うことで各データに対する離散値を求める．この手法ではデータの走査を 2 回行っており，ストリームデータに対するオンライン処理には対応できていない．

2.2 パターンマイニング

パターンマイニングには伊藤ら [2] に提案された，オンライン型アイテム系列パターン抽出アルゴリズムを用いている．これは，Lossy Counting 法 [3] に基づいた ϵ 近似法を拡張したものであり，ストリームデータに対しスライディングウィンドウを用いた頻出系列パターン集合を抽出する．ここで抽出される頻出系列パターン集合を各行動クラスを特徴づける素性とし，行動認識タスクに利用する．

2.3 データ間類似度計算

素性として抽出した系列パターン集合の距離を計算することで，各行動クラスとの類似度を計算する．その際の距離計算には, Jaccard 係数に基づくものを用いている．2 つのパターン集合の共通部分と和集合により演算を行い，x・y・z 軸毎の類似度を算出する．これらを合計することで各行動クラスとの類似度を計算している．

2.4 行動クラス識別

行動クラスの識別には k 近傍法を用いる．これは，入力に対しての類似度が高い k 個のデータを教師データから選び，多数決によってクラスラベルの決定を行うものである．

3. 提案手法

本章では，提案する行動データマイニングのためのオンライン離散化手法について述べる．入力は 3 軸加速度値からなるストリームデータであり，逐次的に離散化処理を行う．本提案手法では，Zhang ら [4] により提案されたオンライン型分位数近似計算手法を応用する．オンラインで近似計算される分位数を離散化時の境界点とし，オンライン離散化を実現する．本章ではまず Zhang らによる先行研究について述べ，次にそれを応用した離散化処理について述べる．その後，提案手法により発生する誤差について述べる．

3.1 オンライン型分位数近似計算手法

本節では，Zhang ら [4] により提案されたオンライン型分位数近似計算手法について述べる．これは，許容誤差の範囲内で入力ストリームデータに対して圧縮を行い，任意の分位率に応じた分位数を近似計算するアルゴリズムである．

入力として，データ長 N のストリームデータ $X = < x_1, x_2, x_3, \cdots, x_N >$ と許容誤差率 ε を与える．ここで，x_i は i 番目に到達した数値データである．出力は，データ長 m のソート済みのデータ列 $X' = < x'_1, x'_2, x'_3, \cdots, x'_m >$ である．$x'i$ は X' 上で i 番目の数値データである．要素数 m は，入力データ長 N と誤差率 ε から算出される．ここで，ユーザ指定の 0〜1 の分位率 ϕ に対し，ϕ に関する分位数 $x'_{\phi m}$ は，$\phi N - \varepsilon N \leq rank_X(x'_{\phi m}) \leq \phi N + \varepsilon N$ を満たす．$rank_X(x_i)$ は，X 上での x_i のランク (上位何番目か) を示す．

なお，平均時間計算量は $O(Nlog(\frac{1}{\epsilon} log\ \epsilon N))$，空間計算量は $O(\frac{1}{\epsilon} log^2(\epsilon N))$ となっている．

階層的データ構造を内部で保持させ，データサイズ N のストリームデータに対し，階層レベル毎にサイズ $b = \lfloor \frac{log \varepsilon N}{\varepsilon} \rfloor$ のデータ容量を持つブロックを用意する．処理手順を以下に示す．手順図を図 1 に示す．

1. 数値データをレベル 0 に読み込む (図 1-[1])
2. レベル 0 のデータがブロックサイズ b だけ溜まった時点で以下の処理を行う
3. レベル 0 内でデータを昇順に並び替える (図 1-[2])
4. ソート後データに対し，偶数番目のデータを削除する (図 1-[3])
5. 削除後，残ったデータをレベル 1 へ転送する (図 1-[4])
6. 上位レベルにおいてもデータが溜まった時点で同様にソート・削除・さらなる上位レベルへの転送処理を再帰的に繰り返す

ここで，一般にストリームデータ長 N は未知である場合が多い．そこで，入力ストリームデータをサブストリーム $P_0, P_1, P_2, \cdots$ に区切り，圧縮処理を行う．サブストリーム P_i のデータ長は $\frac{2^i}{\epsilon}$ であり，データ量がこれを超えるとサブストリームを更新する．分位数を計算する際には，各サブストリームで生成された階層構造すべてを結合・ソート処理を行い，1 つのデータ列 X' を生成する．これは入力ストリームデータ X を圧縮したものであり，任意の分位率に対する分位数を許容誤差率 ϵ の範囲内で近似する．

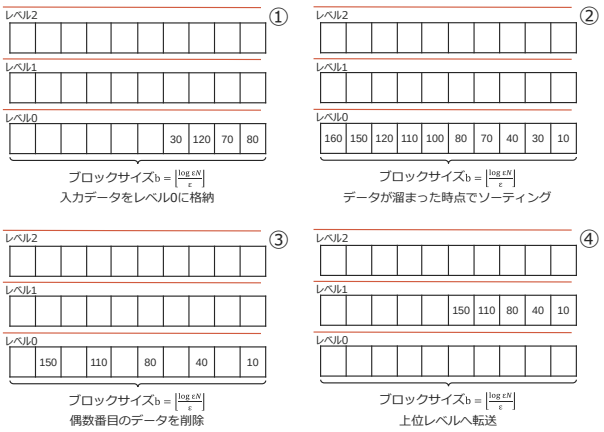

図 1: 圧縮アルゴリズム処理手順

3.2 境界点の導出・離散化処理

前節のアルゴリズムにより生成されたデータ列 X' を用い，離散化のための境界点の集合 Y を求める．入力として，ある時刻 t において生成されたデータ列 X'_t と離散化の際の分割数 d を与える．新規ストリームデータ x_t が入力されたときに，離散化を行うためには，時刻 t での入力済みデータを d 分割する境界点の集合 $Y_t = \{y_1, y_2, y_3, ..., y_{d-1}\}$ を求める必要がある．前節のアルゴリズムは任意の分位率に対して分位数を近似計算するものなので，各 $\frac{1}{d}$, $\frac{2}{d}$, $\frac{3}{d}$, $\cdots$, $\frac{d-1}{d}$ に対する分位数を求めることで，許容誤差の範囲内で入力済データを等深分割法により d 分割することが可能となる．

1. データを読み込む
2. 内部で保持している要約構造を更新する
3. 要約構造から各境界点 y_i を算出する
4. 境界点を用いてセンサ値を離散化する
5. 1 に戻り，以降繰り返す

求まった境界点と，新規ストリームデータの比較を行い，離散化処理を実現する．各加速度データが入力されるたびに圧縮されたデータ列 X' を前節のアルゴリズムを用いて更新し，各境界点 y_i を逐次更新することで離散化処理を実現する．

3.3 提案手法に伴う離散化誤差

提案手法により，オンラインでの離散化処理が実現する一方，2 種類の誤差が発生する．

3.3.1 圧縮処理による境界点の近似誤差

3.1 節で述べたアルゴリズムでの許容誤差率 ϵ により，行動識別に影響を及ぼす．ここで，ϵ が行動識別へ及ぼす影響について以下のことがいえる．

定理 1 (ϵ による最小誤差)
行動識別での誤差を 0 に抑えることはできない．これは，非可逆圧縮を行うため，圧縮時の情報が抜け落ちるためである．

定理 2 (ϵ による最大誤差)
最大誤差は $\frac{\epsilon}{d}$ となる．圧縮処理による誤差幅は ϵ で保障されており，$\epsilon \geq \frac{1}{d}$ を満たすと離散化時に 2 以上の誤差が発生する．$\epsilon < \frac{1}{d}$ を満たすことで，離散化時の最大誤差を 1 に抑えることができる．

3.3.2 ストリーム処理による離散化誤差

オフラインによるバッチ処理ではデータ分布全体を把握した後に境界点を決定できるため，境界点は一意に定めることができる．しかし，オンライン処理では将来的に到達するデータは考慮できず，その時点での境界点を求める．この誤差の影響によっても，行動識別率が低下することが予測される．

4. 検証実験

4.1 使用データ

実験には Hasc2011corpus [5] で収集された 3 軸の加速度データを用いる．「静止」・「歩く」・「走る」・「スキップ」・「階段を上る」・「階段を下りる」の行動クラスからなっている．各データには 19 秒間の加速度センサ値が 10ms 毎に記録されており，それぞれの行動クラスにつき 200 データを用意した．行動識別実験の際には，9 割を学習データ，1 割をテストデータとする交差検定により行動識別性能を評価した．

4.2 実験パラメータ

実験の際のパラメータは次のように設定した．離散化における分割数 4，圧縮アルゴリズムにおいての許容誤差率 0.1，パターンマイニングアルゴリズム中のウィンドウ幅 10，最小サポート値 3，抽出パターン長の下限値 4，k 近傍法における k を 3 と設定する．

4.3 速度検証

性能評価の指標として，速度検証を行った．Android 向けオンライン離散化アプリケーションを開発し，平均遅延時間と最大遅延時間を測定した．テスト用端末には Google 社の Nexus 5X を用いた．

4.4 提案手法での圧縮率

本提案手法の性能評価として，入力ストリームデータをどの程度圧縮できているのか検証を行った．入力されたストリームデータと内部で保持している要約データサイズを測定し，各ストリームデータ長においてどの程度圧縮が実現できているのか検証を行った．

4.5 先行研究との行動識別率の比較

提案手法に基づいて離散化処理した際に，先行研究 [1] と比較してどの程度行動識別率が変化するのか比較検証を行った．先行研究では誤差を含めないオフラインでの離散化を行っている一方，オンライン処理を行う提案手法では 3.3 節で述べたような離散化誤差が発生しており，行動識別率は先行研究のほうが高まることが予測される．圧縮による境界点の近似誤差の影響を測定するために，データを読み込みつつオンライン圧縮を行い，最終的に生成されたデータ列によりオフライン離散化を行ったデータを用いた行動識別率も測定した．これにより，3.3.1 節で述べた圧縮処理による境界点の近似誤差によりどの程度行動識別率に影響を及ぼすのかが把握でき，そこからさらに 3.3.2 節で述べたストリーム処理による離散化誤差による影響度も推定することができると考えられる．

4.6 ストリームデータ長の変化と行動識別率の推移

入力されたストリームデータの変化に応じて，行動識別率がどのように推移するのか検証した．データ量が増えることで学習データが増大するため，行動識別率は増大することが予測される．さらに，データ量の増大に伴い，オンライン離散化時の境界点は次第にオフライン離散化時の境界点に漸近的に近づいていくことが確認されている．データが溜まりきっていない状態では境界点が大きく変動していくが，データが溜まるに従い境界点の変動が小さくなりオフラインでの境界点に収束していく．したがって，データ量の増大に伴いオフラインとオンラインによる離散化誤差の影響は小さくなっていくと予想され，先行研究 [1] との行動識別率の推移の比較も検証する．

5. 実験結果

5.1 速度検証

Android アプリを動作させた際の処理時間の推移を図 2 に示す．1 時間アプリを動作させたところ，平均遅延時間は 0.55ms，最大遅延時間は 52ms となった．10ms 毎の動作を期待しているため，遅延による影響は大きいものといえる．図 2 のグラフから，常に遅延が生じているわけではなく，内部で保持している要約構造が大きく更新された際に大きな遅延が生じていることがわかる．ここから，遅延が発生しているタイミングでは加速度センサの値をバッファに一時格納し，要約構造の更新が終わった時点でバッファ中に溜まったデータを逐次処理していく手法をとることで，遅延による影響は軽減できるのではないかと考えられる．

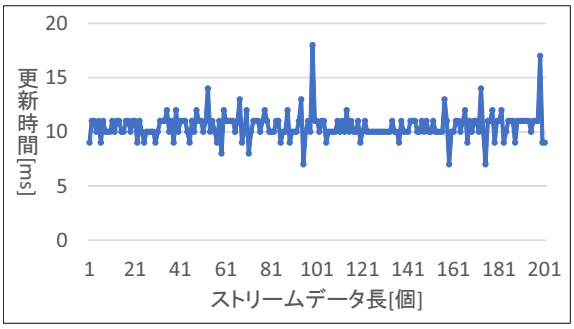

図 2: アプリ動作値の処理時間の推移

5.2 提案手法での圧縮率

ストリームデータを入力した際の要約後のデータサイズの推移を図 3 に示す．図 3 から，入力ストリームデータを最大で 99.8%圧縮できており，この程度のデータ量であればメモリ上に要約構造を常に保持しながらオンライン離散化を行うことは十分可能であるといえる．

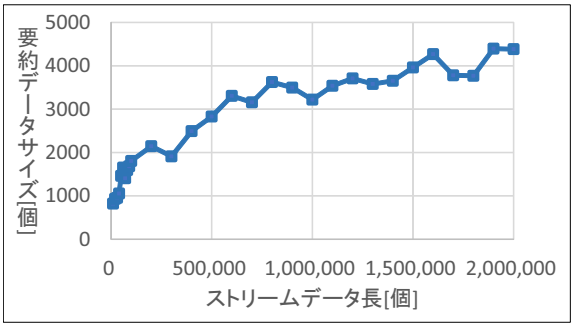

図 3: 提案手法での圧縮率

5.3 先行研究との行動識別率の比較

先行研究と提案手法との行動識別率の比較実験の結果を図 4 に示す．図 4 のように，先行研究 [1] では 83%の識別率を記録しており，提案手法の識別率は 77%となった．オンライン離散化処理実現のために誤差を許容した結果，識別率が低下していることがわかる．また，最終的に生成された要約構造により求まる境界点を用いて，オフラインで離散化処理を施して行動識別した結果，識別率は 80%となった．これらの結果を踏まえ，3.3 節で述べた提案手法においての 2 つの誤差による識別率への影響を考える．3.3.1 節で述べた圧縮処理による境界点の近似誤差による識別率への影響は，$83\% - 80\% = 3\%$ 程度であると考えられる．さらに，3.3.2 節で述べたストリーム処理による離散化誤差については，$80\% - 77\% = 3\%$ 程度であると考えられる．

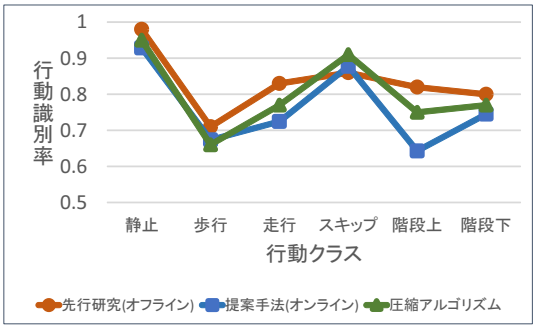

図 4: 先行研究との行動識別率の比較

5.4 ストリームデータ長の変化と行動識別率の推移

ストリームデータ長を変化させた際の行動識別精度の推移のグラフを図 5 に示す．ストリームデータ長が 520,000 の時点での識別率と 2,280,000 での識別率の差を先行研究との比較検証を行った結果，T 検定の P 値は 0.000123 となり有意な差があるといえる．ここから，データ量の増大に伴い，オンライン離散化の際の境界点がオフライン離散化の境界点に近づいていくことで，行動識別率も漸近的に近づいていくことがわかる．

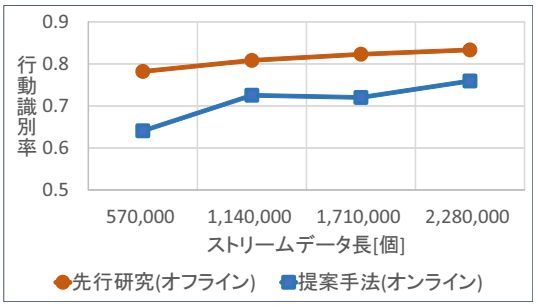

図 5: ストリームデータ長の変化と行動識別率の推移

6. 今後の課題

6.1 識別結果の考察

今回，行動識別率に焦点を当てて結果を検証実験を行っており，誤識別の場合にどの行動クラスに識別されているのかまでは検証を行っていない．さらなる識別率の向上や，人間行動の理解のためには，どの行動クラスに誤識別されているのかの検証を行い，それを踏まえて手法を見直す必要がある．

6.2 特徴量の追加

今回，実験データとして用いた HASC データには，10ms 毎の 3 軸の加速度センサ値が記録されている．今回は実験データとして加速度センサ値のみを対象とし，離散化・パターンを抽出することで行動識別を行った．一方，先行研究では 3 軸加速度値から重力ベクトルといった特徴ベクトルを算出し，それらも特徴量として扱うことで行動識別率が増加することが確認されている [7][8]．本研究でもこの発見を活かすことができ，オンラインで新たな特徴量の計算を行うことで，さらなる行動識別率の向上が狙えることが予測されるため，離散化処理部に特徴量も追加を行う機能を実装したいと考えている．

7. おわりに

本研究では，行動データマイニングアルゴリズム中の離散化処理部分において，誤差を許容することでオンライン処理を実現するアルゴリズムを提案した．入力ストリームデータに対し，許容誤差の範囲内で圧縮を行うことで，内部で保持すべきデータ数を大幅に減少させることができた．

先行研究との行動識別率の比較検証実験の結果，99.8%の圧縮率を実現しつつ，行動識別率 77%を記録した．誤差を許容することで認識精度は多少劣ってしまったものの，データ量の増加に伴い，識別率は漸近的に先行研究に近づいていくことを示すことができた．

謝辞

本研究は一部，ISPS 科学研究費補助金（No.25330256）および JST さきがけの援助を受けている．

参考文献

[1] 林悠樹: オンラインマイニングによる行動認識パターンの抽出, 2014 年度山梨大学工学部コンピュータ・メディア工学科卒業論文 (2015)

[2] 伊藤秀志, 岩沼宏治, 山本 泰生: バースト出現へ対応を目的としたオンライン型系列マイニングへのメモリ制限の導入, SIG-DOCMAS: データ指向構成マイニングとシミュレーション研究会 (2011)

[3] Gurmeet Singh Manku, Rajeev Motwani.: Approximate Frequency Counts over Data Streams, International Conference on Very Large Data Bases (2002)

[4] Zhang Qi, Wang Wei.: A fast algorithm for approximate quantiles in high speed data streams, Proceedings of the 19th International Conferende on Scientific and Statistical Database Management, pp. 29, July 09-11 (2007)

[5] Human Activity Sensing Consortium Challange, 〈 http://hasc.jp/ 〉 (2016-2-1).

[6] 大内一成, 土井美和子: スマートフォンを用いた生活行動認識技術, 東芝レビュー Vol.68 (2013)

[7] 松重龍之介, 角所孝, 岡留剛: 半教師あり RVM による加速度データからの行動推定, 第 28 回人工知能学会全国大会 pp. 1-4 (2014)

[8] Ling Bao, Stephen S. Intille.: Activity Recognition from User-Annotated Acceleration Data, Volume 3001 of the series Lecture Notes in Computer Science, pp. 1-17, (2004)