

# スマートデバイスより取得した センサデータを用いた歩行時の生体認証

吉丸憂彌 (佐賀大学大学院先進健康科学研究科) †

堂菌浩 (佐賀大学理工学科電気電子工学部門) ‡

## 1. はじめに

現在、スマートフォンをはじめとしたスマートデバイスが世界中で普及している。それに伴い、スマートデバイスを用いた各種サービスの提供が多数行われ、私達の生活は日々便利なものとなっている。しかし、そのサービスを利用するためには、スマートデバイスにクレジットカード情報や銀行口座番号等の重要な個人情報を紐づける必要がある。[1]また、スマートホームやデジタルキー等のサービスが普及してきており、情報のみでなく、電化製品や車といったモノまでもがスマートデバイスに紐づけられている状況である。[2]このように、現在スマートデバイスには様々な情報やモノが紐づけられており、これまで以上のセキュリティ強化が求められている。そこで、スマートデバイスの所持者が、本来の所有者かどうかを常時判別するシステムを開発したいと考えた。その判別方法の1つとして、歩行時に生体認証を行うことを提案する。1日平均男性8,202歩、女性7,282歩という日常的に人が行っている歩行動作は、目標とする判別システムに適しているのではないかと考える。[3]

## 2. 研究目的

本論文における最終目標は、スマートフォンの使用者が本来の所有者かどうかを常時判別するためのシステムに用いる、歩行時の行動的生体認証の精度向上である。スマートフォンに搭載されたセンサデータを用いた行動的生体認証には、3つの段階がある。1つ目が、所持者がどのような行動を行っているかを推定する状態推定。2つ目が、スマートフォンが所持されている位置を推定する所持位置推定。最後に本来の所有者かどうかを判別するユーザー推定である。[4]今回は、状態推定が歩行時、所持位置推定がズボンの右ポケットという限定的条件下で実験を行う。また、ユーザー推定は、教師なし学習を用いた外れ値検知という観点から判別する。理由としては、1台のスマートフォンに対して1人の所有者であることが一般的であるため、歩行時のセンサデータはその所有者からしか取得することが出来ない。そのため、1人分のセンサデータからユーザー推定が可能であることが望ましいと考えたためである。

## 3. 実験方法

### 3.1 センサデータの取得

実験用端末として、OPPO社のCPH1943を使用し、この端末に搭載されている3軸加速度センサと3軸ジャイロセンサよりセンサデータを取得した。データを取得するために、PythonにおけるGUIアプリ開発のためのフレームワークであるkivyを用いて、kivyアプリを開発した。その後Buildozerと呼ばれるツールを用いて、パッケージ化した。このアプリでは、データ取得の際のサンプリング周波数を50Hzと設定した。

歩行時のセンサデータを取得するために、男性4人を被験者とした。以降この4人の被験者をA, B, C, Dと呼ぶ。アプリを起動した状態で、それぞれの被験者のズボンの右ポケットに端末を所持し、7分間の歩行を行った。

## 3.2 特徴量の取得

3.1で取得した歩行時の3軸加速度と3軸角速度をパソコンに転送し、特徴量を取得するための処理を行った。まず、端末の向きを考慮しなくて良いように、3軸加速度、3軸角速度それぞれから、3軸合成加速度、3軸合成角速度を求めた。そのセンサデータの波形を下の図1に示す。

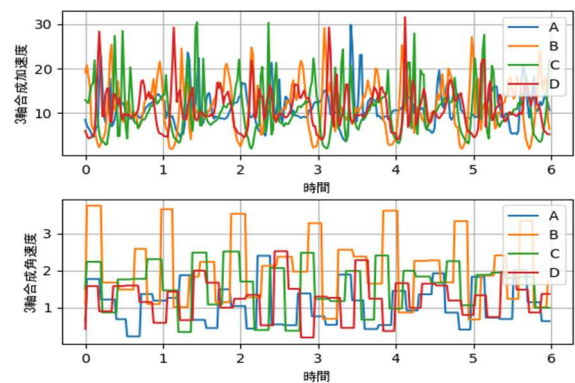


図1. 三軸合成加速度と三軸合成角速度の波形

上記の図1より、歩行時に取得できるセンサデータにはある一定の周期が存在していることが分かる。この3軸合成加速度、3軸合成角速度から個人を識別できる特徴量を取得するための方法として、以下の4種類の処理を行い精度の比較をした。

- (1) データを50個ずつに分割する。
- (2) 単純移動平均によりノイズを軽減した後、データを50個ずつに分割する。
- (3) FFTにより振幅スペクトルを算出する。
- (4) FFTにより振幅スペクトルを算出した後、単純移動平均によりノイズを軽減する。

(1)の方法では、まず加速度と角速度では単位が異なるため、標準化を施し尺度を揃えた。その後、データを50個ずつに分割し、その50個のデータを1つの入力データとした。また、分割する際のフレームを0.2秒ずつずらし、前後の入力データに49個の重複があるようにした。そうすることで、少ないデータから多くの入力データパターンを作成することが出来た。取得したセンサデータ4分を訓練データ、2分をテストデータとした。

(2)の方法では、分割方法等は(1)と同様に行うが、ノイズを軽減した際に判別精度に変化が現れるかを調べる目的で、データ分割を行う前に参照数5で単純移動平均を行った。

(3)の方法では、標準化を行うこと以外(1)と同じ要領でデータを64個ずつに分割した。そして、周期性を持たせるためにハン窓を用いてウィンドウ処理を行った。その後FFTを行い、ナイキスト周波数25Hz以上の周波数を切り捨て、振幅スペクトルを算出し入力データとした。また、約4分を訓練データ、約2分をテストデータとした。

(4)の方法では、(3)と同様に振幅スペクトルを算出した

後、参照数 5 で単純移動平均を行いノイズの軽減を行った。

以上の方法で3軸合成加速度、3軸合成角速度から特徴量を取得する。分類器には、3軸合成加速度から得た特徴量を入力とする場合と、3軸合成角速度から得た特徴量を入力とする場合、その両方を入力とする場合で精度を比較する。

### 3.3 分類方法

ある被験者が本来の所有者であるとした場合、それ以外の被験者を外れ値とし判別を行った。また、本来の所有者がAであると仮定した場合、Aの訓練データを用いて分類器の学習を行った。その後、Aのテストデータを用いて正常値が分類できているか、その他の被験者のテストデータを用いて異常値が分類できているかという検証を行った。今回の研究で用いた分類器は、以下の2つである。Scikit-learnを用いて実装した。

(1) OCSVM(One Class Support Vector Machine)

(2) IsolationForest

(1)のOCSVMは教師なし学習の一種で、新しいデータを学習セットと類似または異なるものとして分類することが出来る。[5]パラメータにkernel, gamma, nu等があるが、今回kernelはrbf, gammaはscaleを選択した。nuの値を変化させてグラフを作成した。

(2)のIsolationForestは、決定木を用いた教師なし学習の一種である。異常なデータはすぐに分割されてしまうという特性を生かして異常検知を行う。[6]contaminationというパラメータを変化させ、精度に関するグラフを作成した。

### 4. 実験結果

作成したグラフの中でもっとも精度が高かったものを下の図2に示す。

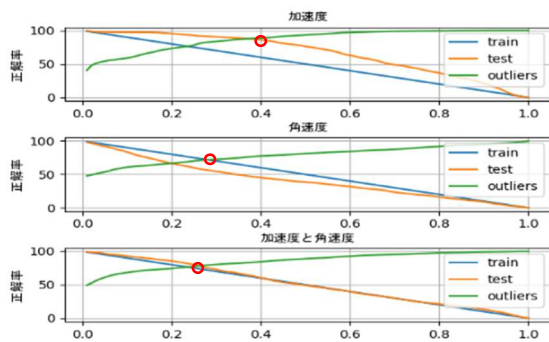


図2、平滑化を行った後、データを50個ずつに分割した場合の正解率  
次に、分類器 OCSVM を用いた際の、各処理におけるクロスポイント下での精度を下の表1に示す。

表1、本来の所有者A、分類器 OCSVM 時の各処理におけるクロスポイント下の精度(%)

	加 速 度			角 速 度			加速度&角速度		
	訓練	テスト	外れ値	訓練	テスト	外れ値	訓練	テスト	外れ値
分割	52.00	75.65	75.97	81.99	66.13	65.00	69.98	74.45	74.13
分割 (平滑化)	63.02	87.83	87.72	72.03	56.35	71.74	74.03	78.85	78.13
FFT	46.00	50.06	48.05	43.01	46.63	52.48	49.93	48.30	49.98
FFT (平滑化)	47.00	49.43	48.48	46.01	50.38	50.16	47.00	49.58	48.32

次に、分類器 IsolationForest を用いた際の、各処理におけるクロスポイント下での精度を表2に示す。

表2、本来の所有者A、分類器 IsolationForest 時の各処理におけるクロスポイント下での精度(%)

	加 速 度			角 速 度			加速度&角速度		
	訓練	テスト	外れ値	訓練	テスト	外れ値	訓練	テスト	外れ値
分割	64.90	80.90	67.76	80.20	63.67	67.37	77.60	73.30	75.06
分割 (平滑化)	69.80	87.60	69.82	79.40	63.75	63.16	75.90	74.90	75.85
FFT	50.00	57.39	43.59	50.00	55.16	47.54	50.00	55.54	47.28
FFT (平滑化)	50.27	53.60	45.38	50.00	53.64	48.04	50.04	53.02	47.19

### 5. 考察

nu と contamination は訓練データにどれだけ異常データが含まれているかを示すものである。[5][6]それらのパラメータ値を大きくすることによって、異常に関して敏感になる。同時に、正常値を異常だと判断し易くなる。今回はセキュリティを目的としているので、正常値を異常であると誤分類したとしても、異常値に敏感になるよう各パラメータを設定した。そして、図2の赤く囲まれたクロスポイントを適切なパラメータとして選択した。表1、表2より、加速度、角速度両方の値を用いた方が訓練データ、テストデータ、外れ値の全てにおいてバランスの良い精度となることが分かる。また、今回用いた振幅スペクトルはどちらの分類器においても、全てにおいて精度が50%以下となり、きちんと特徴量が取得できていないことが分かる。サンプリング周波数を大きくする、またはより大きなフレームで分割しFFTを行う等の改善が必要である。データを分割したものは、ある程度の精度で分類でき、ある程度特徴量が取得できていると考えられる。今回の結果からは、OCSVMはノイズを除去すると精度が高まり、IsolationForestはノイズを含んでいてもある程度精度を出すことが出来る分類器だと分かった。

### 6. まとめ

今回は nu と contamination のみにおいてパラメータチューニングを行った。次回は、これら以外のパラメータチューニングを行い、精度向上を目指したい。

### 7. 参考文献

[1]PayPay(ペイペ)- QR コードで支払うキャッシュレス決済のスマホアプリ  
<[https://www.mhlw.go.jp/wwwl/topics/kenko21\\_11/b2.html](https://www.mhlw.go.jp/wwwl/topics/kenko21_11/b2.html)>  
[2]デジタルキーとは | デジタルキー | トヨタ自動車WEBサイト  
<[https://toyota.jp/digital\\_key/details.html](https://toyota.jp/digital_key/details.html)>  
[3]健康日本 21(身体活動・運動) | 厚生労働省  
<[https://www.mhlw.go.jp/wwwl/topics/kenko21\\_11/b2.html](https://www.mhlw.go.jp/wwwl/topics/kenko21_11/b2.html)>  
[4]岩本健嗣, 杉森大輔, 松本三千人:3軸加速度センサを用いた歩行者推定法, 情報処理学会論文誌 vol.55 No.2 739-749(Feb, 2014)  
[5]sklearn.svm.OneClassSVM<sklearn.svm.OneClassSVM-scikit-learn1.0.2 documentation>  
[6]sklearn.ensemble.IsolationForest<sklearn.ensemble.IsolationForest-scikit-learn1.0.2 documentation>