

スマートウォッチデータからのイベント検出のための 多重時間スケール解析手法の検討

佐藤 哲†

パーソルキャリア株式会社†

1. はじめに

安価なスマートウォッチの普及により、睡眠データや心拍データなどの生体データが医療機器を用いなくても簡単に取得できるようになってきている。しかし、安静な状態で医療機器により測定する場合と比較すると、データの正確性などの点で利用が難しいことが知られている。その理由としてはセンサーの特性とアーチファクトの存在があげられるが、いずれも扱いが難しい反面、医療現場では得られないデータが含まれると考えられるため、データ分析手法を工夫して効果的に利用することが求められている。そこで本研究では、ノイズに強い複数の手法の導入と、人間の生活に内在する複数の時間的周期を利用することで、スマートウォッチから得られる生体データを効果的に分析する手法を提案する。本研究は、スマートウォッチから得られる生体データの一つである心拍データから、人間がどのような活動をしているかというイベントを検出することを目標とする。

2. スマートウォッチにより収集される生体データの分析

スマートウォッチは、光学式心拍センサ、皮膚温センサ、加速度センサ、GPS やジャイロスコープなどの各種センサが内蔵されている。そのため日常生活の中で生体データを容易に収集可能である一方、収集されたデータを分析し意味のある情報を取り出すためには、2つの課題がある。

まず、使用されているセンサが医療機器に使用されているセンサと特性が違う問題がある。本研究で対象とする心拍データの場合、医学的には心臓の電気信号を記録する心電図 (Electrocardiogram; ECG) を利用することが一般的である。一方で、心拍に関するデータを測定するためにスマートウォッチで用いられているセンサは光電脈波記録 (Photoplethysmography; PPG) であり、皮膚表面の血管に光を当て、反射光の強さを測定することで血流を測定する。確立された手法である ECG に比べ PPG は精度が低いとされ、PPG の精度向上のための研究が続いている [1]。精度が低いとされる主な理由としては、ECG が電位の測定により心拍の変動を直接的に測定できるのに対し、PPG は血流の変動から間接的に心拍

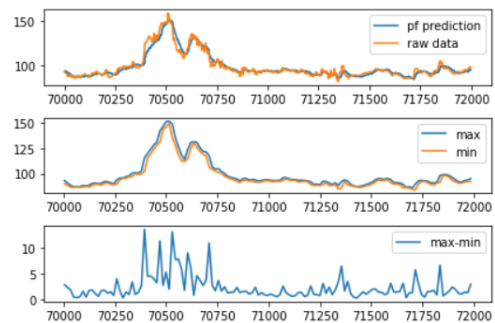


図 1: ダウンサンプリング例

の変動を測定しなければならないことによる。次に、ECG も PPG もアーチファクト (ノイズ) の影響を受けることは変わりがないが、ECG が比較的安静時の測定に使用されるのに比べ PPG は運動時の測定に使用されることも多く、よりアーチファクトの影響が大きいとされる [2]。また、ECG は専門家が決まった手続きにより測定し分析して結果を出すことに対し、PPG は専門家の手続きを介さないという差がある。そのため、アーチファクトの影響を除外するか、アーチファクト自体の情報を利用することができれば、ECG では得られない情報が得られる可能性もある。

このような特徴がある PPG データに対しては、ECG のデータ分析手法を援用する方法 (例えば [3]) と、PPG データと得たいデータを end-to-end で対応付ける方法 (例えば [4]) の 2通りがある。本研究のアプローチは後者であり、前者のアプローチに必要な、例えば入力信号から、ECG のデータ分析の基本となる RR 間隔 (心電図のピークとピークの間隔で、1分辺りの心拍数の 1/60 に近い値となる) を推定するような処理は行わずに、PPG データと、そのデータが取得された時にユーザがどのような活動をしていたかというイベントを対応付けることを目標とする。

3. 分析手法

3.1. 複数の時間スケール情報の考慮

本研究で用いているデバイスからは秒単位のデータを得ることができるが、イベント検出が目的であることを考えると、数秒~数十秒単位での変化が検知できることが望ましい。また、現実社会での人間の生活の中で 24 時間の周期は重要であるので、24 時間単位での変化も考慮したい。そこで本研究では、

Study of Multi-timescale Analysis for Event Detection from Smartwatch Sensor Data

†Tetsu R. Satoh, PERSOL CAREER CO., LTD.

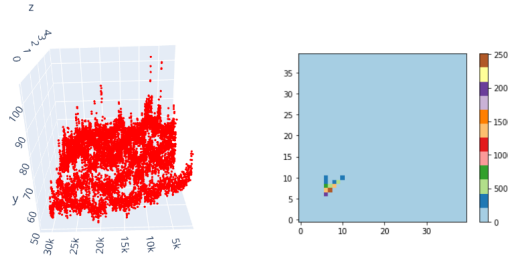


図 2: 3次元データとパーシステンスダイアグラム変換の例

粒子フィルタにより欠損値補間・平滑化を行った後、 t_w 秒の時間窓で入力信号をダウンサンプリングすることで数十秒単位での時間スケールに対応する。ただし、単純に平均値や中央値を代表値とすると情報の欠落があり得るため、最大値と最小値を保持する。時間窓の中で急激な信号の変化があった場合は最大値と最小値の差が大きくなるため、最大値と最小値は変化点の情報を保持する。図 1 に、ダウンサンプリング結果の例を示す。図は上から順に、加工しないオリジナルデータ及び粒子フィルタによる推測結果データ、粒子フィルタによる推測結果を離散化し最大値と最小値を求めた結果、最大値と最小値の差を示す。また、24 時間単位での時間スケールに対応するためには、時間-入力信号の x-y 軸 2 次元データを 24 時間単位で z 軸上に並べた 3 次元データを構成する。3 次元データを構成した後、3 次元の点群データはスパースでデータサイズが大きく解析が難しいため、パーシステンスダイアグラム変換により 2 次元のベクトルに変換する。図 2 に、構成した 3 次元データとそのパーシステンスダイアグラム変換の例を示す。以上、処理に必要な粒子フィルタとパーシステンスダイアグラム変換については、次節以降で説明する。

3.2. 粒子フィルタ

PPG 信号に対しては、サンプリングレートの低さやアーチファクトによるノイズに対処するために、スプライン補間 [5]、主成分分析 [2]、離散ウェーブレット変換 [4] など様々な手法が導入されている。本研究では、欠損値補間や平滑化のために粒子フィルタ [6] を利用する。

粒子フィルタのアルゴリズムは以下である。次のようなシステムモデルと観測モデルを考える：

$$\begin{cases} x_k = F(x_{k-1}, v_k) \\ y_k = H(x_k, \omega_k) \end{cases} \quad (1)$$

ここで、 x_k 及び y_k は、時刻 k での状態ベクトル及び観測ベクトル、 v_k 及び ω_k はシステムノイズ及び観測ノイズと呼ばれる量である。状態ベクトルは観測できない隠れ変数であり、1 ステップ前の状態ベクトルと観測ノイズによって決まると仮定し、観測可能な観測ベクトルは状態ベクトル及び観測ノイズによって決まると仮定する。このモデルに対し、次

のアルゴリズムにより、観測ベクトルから状態ベクトルを推定する：

- (1) システムモデルに基づき状態ベクトルを予測する。

$$p_k^{(i)} = F(f_{k-1}^{(i)}, v_k)$$

ここで、 f_{k-1} は 1 ステップ前の状態ベクトルを近似する粒子ベクトルである。

- (2) 観測値 y_k を得る。
- (3) 予測値 $p_k^{(i)}$ と観測値 y_k より、観測モデルを用いて予測値の信頼度を表す尤度 $\alpha_k^{(i)}$ を求める：

$$\alpha_k^{(i)} = r(H^{-1}(y_k, p_k^{(i)})) |\partial H^{-1} / \partial y_k|$$

ここで、 r は観測ノイズの確率密度関数であり、 H^{-1} は関数 H の逆関数で状態ベクトル x_k と観測ベクトル y_k の関数であるが、状態ベクトルの代わりに予測値である $p_k^{(i)}$ を代入する。

- (4) 粒子の尤度 α_k に基づき、粒子 p_k を再サンプリングし状態ベクトルを近似する粒子ベクトル f_k を求める：

$$f_k^{(i)} = \text{確率 } \alpha_k^{(j)} / \sum_{l=0}^{n-1} \alpha_k^{(l)} \text{ で } p_k^{(j)}, 0 \leq j < n$$

ここで、 n は粒子の個数である。

- (5) 粒子群 $f_k^{(0)}, f_k^{(1)}, \dots$ の期待値等を求め、状態ベクトルの推定値とする。

粒子フィルタは、状態ベクトル x_k を、状態ベクトルが取る可能性のある複数のベクトルを粒子ベクトル $f_k^{(i)}$ として保持し、状態ベクトルの位置を確率で表現することが特徴である。本研究での観測ベクトルはスマートウォッチから得られる心拍数であり、ノイズが含まれているスマートウォッチから得られるデータから真の値を推定するために粒子フィルタを利用している。

3.3. パーシステンスダイアグラム変換

本研究では、24 時間の生体信号を 1 日単位で並べた 3 次元データを構成する。このようなノイズが含まれる 3 次元データから特徴を抽出し、2 次元に次元を削減する手法としては、パーシステンスダイアグラム変換が有効である。パーシステンスダイアグラム変換 (Persistence Diagram Transformation; PDT) は、任意の次元のデータを 2 次元のベクトルデータ列に変換する手法で、ノイズに対しロバストであり、かつ効果的な特徴抽出が可能となる手法である [7]。

PDT のアルゴリズムは以下である：

- (1) 各データ x_i に対し、データの次元 n の空間の中でデータを中心とする半径 r 球 B_r を配置する

$$B_r(x_i) = \{x \in R^n \mid \|x - x_i\| < r\}$$

- (2) 半径 r を増大させ、球同士の交差点により発生する位相的特徴量の穴を抽出し、穴の発生時刻と消滅時刻を記録する (フィルトレーション, 図 3)

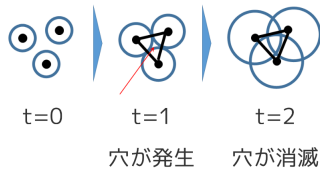


図 3: フィルトレーションの例

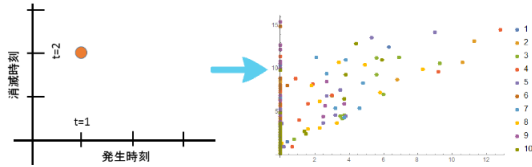


図 4: パーシステンス図の例

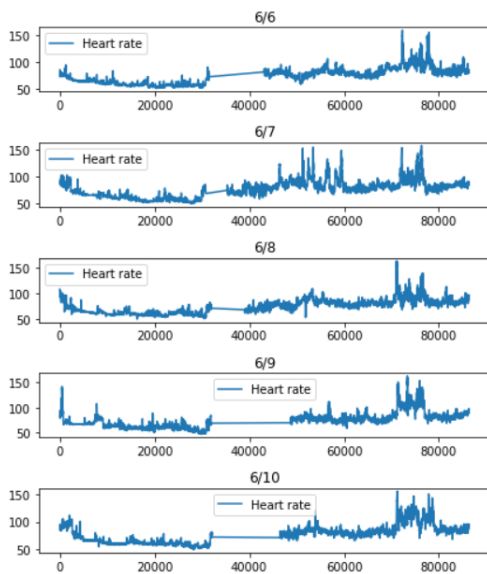


図 5: テスト用心拍データ

- (3) 発生時刻を x 座標, 消滅時刻を y 座標とするパーシステンス図を生成し, 2次元ベクトル群とする (図 4)

パーシステンスダイアグラム変換は, 幾何的な特徴である穴の発生と消滅を検出する手法であり, スケールを問わずにデータの変化点を抽出することが可能である. 本研究では, 数十秒のスケールでの心拍データの変動を検出すると共に, 24時間スケールでの変動 (例えば, ある時間帯に平日は運動しているが, 休日は運動していないなど) を捉えることに利用している.

4. 実験

簡単な例として, 判別が容易なイベントとして睡眠時, 運動時, デスクワーク時という3つのイベントを比較した例を紹介する. データを収集するスマートウォッチは, fitbit 社の fitbit sense を用いた[†]. 図

[†]<https://www.fitbit.com/global/jp/products/smartwatches/sense>

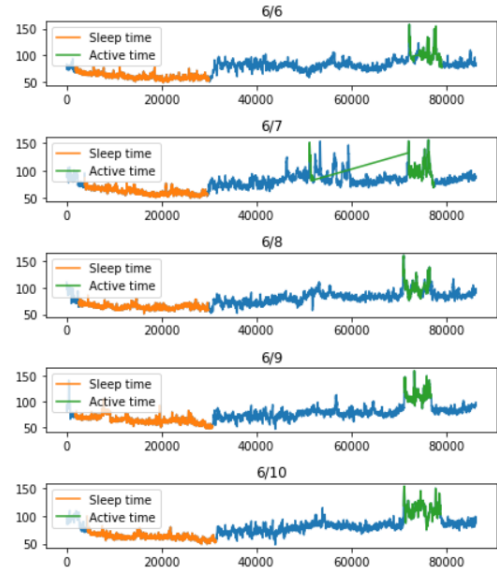


図 6: テスト用前処理後データ

5に, 6月6日から6月10日の測定された心拍データを示す. 縦軸は1分辺りの心拍数であり, 横軸は0時0分からの秒数である. また, 図6に, 図5のデータに対し粒子フィルタをかけ, 平滑化・欠損値補間及びダウンサンプリングを実施し, 睡眠時データと運動時データを抽出した結果を示す. ダウンサンプリングの時間窓の大きさは, 従来研究では4秒[8], 10秒[4]あるいは30秒[9]が用いられていたことを考慮し, $t_w = 15$ とした. ダウンサンプリングは時間窓の中での最大値と最小値を抽出しているが, 本節では簡単のため最大値のデータのみを使用している. 粒子フィルタのモデルは, システムモデル, 観測モデルともに状態ベクトル及び観測ベクトルとノイズの線形和であると仮定し, システムノイズ, 観測ノイズ共にガウス分布に従うとし, 粒子数は20とした. すなわち状態空間モデルは以下である:

$$\begin{cases} x_k = x_{k-1} + v_k, v_k \sim N(\mu, \sigma^2) \\ y_k = x_k + \omega_k, \omega_k \sim N(\mu, \sigma^2) \end{cases} \quad (2)$$

睡眠時と運動時のイベントの定義としては, fitbit 睡眠時データ取得 Web API^{††} 及び fitbit 運動時データ取得 Web API^{†††}により得られるデータを利用した. 睡眠時データは fitbit 社が提供するアルゴリズムに基づいているが, 他のスマートホンアプリのデータにより, 信頼性が高いことを検証している. また運動時データは, 運動の開始と終了をユーザが記録する方式であるため, 信頼性が高いとみなしている. デスクワーク時データは, 睡眠時データ及び運動時データとされている時間帯以外の時間を対象としている. このデータ系列族の中から, 睡眠時データと

^{††}<https://dev.fitbit.com/build/reference/web-api/sleep/>

^{†††}<https://dev.fitbit.com/build/reference/web-api/activity/>

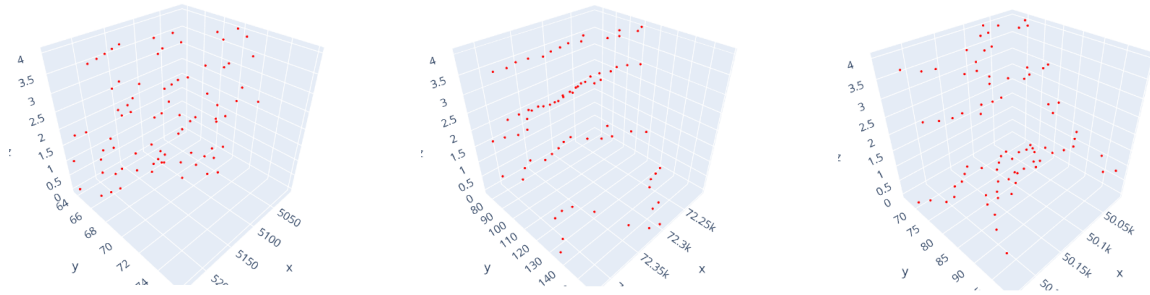


図 7: 左: 睡眠時データ, 中央: 運動時データ, 右: デスクワーク時データ

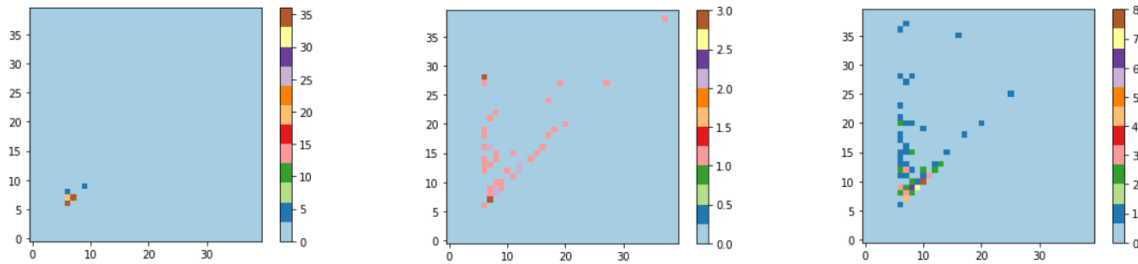


図 8: 左: 睡眠時データ, 中央: 運動時データ, 右: デスクワーク時データ

して 5000 秒～5240 秒 (約 1 時 23 分～1 時 27 分) のデータ, 運動時データとして 72195 秒～72435 秒 (約 20 時 3 分～20 時 7 分) のデータ, デスクワーク時データとして, 50000 秒～50120 秒 (約 13 時 53 分～13 時 57 分) のデータを抽出した結果を図 7 に示す. 図 7 は 3 次元点群データのため判断が付きにくい, 直感的にはデスクワーク時データは, 睡眠時データよりは運動時データに近いことが期待される. これらのデータに対しパーシステンスダイアグラム変換をしたデータを図 8 に示す. 図より, 運動時データとデスクワーク時データは似た三角状の広がりを見せているが, 睡眠時データは左下にプロットが集中していることが分かる. ここでは示さないが, パーシステンスダイアグラム変換の結果はベクトルであるので内積により類似度が定義でき, 近さは数値で示すことができ, カーネル法により効率的な計算が可能である [10].

この実験の結果により, 睡眠時, 運動時, デスクワーク時といったシンプルなイベントは, 心拍データのみから識別できる可能性が示唆されたと言える.

5. おわりに

本研究では, スマートウォッチから得られる生体データの一つである心拍データに対し, 人間の行動をイベントと定義し, 心拍データからイベントを推定し検出する手法を検討した. そして実験により, 粒子フィルタとパーシステンスダイアグラム変換の結果から, 心拍データのみからイベントを検出できる可能性を示唆する結果を得た. 今後は, ニューラルネットワークなどを用いた教師あり学習により分類器を構成し, 心拍データからイベントを識別する手法を検討する予定である.

参考文献

- [1] G. Lu, F. Yang, J. A. Taylor and J. F. Stein, A comparison of photoplethysmography and ECG recording to analyse heart rate variability in healthy subjects, *J. Med. Eng. Technol.*, Vol. 33, pp. 634-64, 2009.
- [2] R. Mao, M. Twardy, S. W. Wegerich, C. J. Goergen, G. R. Wodicka F. Zhu, Motion Artifact Reduction In Photoplethysmography For Reliable Signal Selection, *arxiv.2109.02755*, 2021.
- [3] A. Panda, S. Pinisetty and P. Roop, Runtime Monitoring and Statistical Approaches for Correlation Analysis of ECG and PPG, *arXiv:2202.00559*, 2022.
- [4] Rishi Vardhan K, Vedanth S, Poojah G, Abhishek K, Nitish Kumar M, Vineeth Vijayaraghavan, BP-Net: Efficient Deep Learning for Continuous Arterial Blood Pressure Estimation using Photoplethysmogram, *arxiv.2111.14558*, 2021.
- [5] 吉田 豊, 湯田 恵美, 早野 順一郎, 腕時計型脈波計を用いた睡眠時の心拍変動推定アルゴリズムの開発, *FIT2017*, Vol. 2, pp 339-343, G-012, 2017.
- [6] 北川源四朗, モンテカルロ・フィルタおよび平滑化について, *統計数理*, Vol. 44, No. 1, pp. 31-48, 1996.
- [7] 平岡裕章, 位相的データ解析とパーシステントホモロジー, *数学*, Vol. 68, pp. 361-380, 2016.
- [8] E. Lan, Performer: A Novel PPG to ECG Reconstruction Transformer For a Digital Biomarker of Cardiovascular Disease Detection, *arxiv.2204.11795*, 2022.
- [9] K. Kotzen, P. H. Charlton, S. Salabi, L. Amar, A. Landesberg and J. A. Behar, SleepPPG-Net: a deep learning algorithm for robust sleep staging from continuous photoplethysmography, *arXiv:2202.05735*, 2022.
- [10] G. Kusano, K. Fukumizu and Y. Hiraoka, Kernel method for persistence diagrams via kernel embedding and weight factor, *arXiv:1706.03472*, 2017.