

# ウェアラブル端末と心理入力アプリデータを統合した感情予測アルゴリズムの開発

2025-06-13 進捗報告

森雄大

## 1. T

### 研究の目的

目的は、ウェアラブル端末だけでリアルタイムにネガティブ／ポジティブの感情を高精度予測できるアルゴリズムを構築し、セルフケアに役立つ仕組みを提供すること

### Todo

- ・ 生体データが欠落しているタイムスタンプ情報を除外し、生体データとタイムスタンプで一对一の対応関係をつける
- ・ 被験者ごとの計測回数の偏り
  - － ヒストグラムで被験者ごとの合計計測回数を可視化
  - － → 大きく偏る被験者がいればサブサンプリングや重み付けで是正する必要がある
- ・ 被験者 × 時間帯の測定タイミング偏り
  - － 縦軸＝被験者、横軸＝時間ビンのヒートマップを作成
  - －  $\chi^2$  独立性検定で「被験者と時間帯は独立（一様分布）」を検証
  - － →  $p < 0.05$  なら時間帯に偏りあり → ビンの結合やデータ補正を検討
- ・ 被験者間パラメータ差の有無
  - － 被験者ごとのパラメータ分布をヒートマップで可視化
  - － 次の2つのモデルを比べて尤度比検定（LRT）を行う
    - \* 被験者ごとに同じ切片だけを持つモデル
    - \* 被験者ごとに異なる切片（ランダム切片）を持つモデル
  - － ランダム切片の分散成分が有意（ $p < 0.05$ ）なら「被験者間に偏りあり」→ 共変量化や集約で補正
- ・ 時間帯ビン間パラメータ差の有無
  - － 横軸＝時間ビン、縦軸＝パラメータ値のヒートマップで可視化
  - － 次の2つのモデルを比べて尤度比検定を行う
    - \* 被験者ごとの切片ずれのみを考慮したモデル
    - \* 上記に「時間帯」を説明変数として加えたモデル
  - －  $p < 0.05$  なら「時間帯でパラメータに差あり」→ モデルへの組み込みやビン再設定を検討
- ・ 曜日間パラメータ差の有無
  - － 横軸＝曜日、縦軸＝パラメータ値のヒートマップ作成
  - － 次の2つのモデルを比べて尤度比検定を行う
    - \* 被験者ごとの切片ずれのみを考慮したモデル
    - \* 上記に「曜日」を説明変数として加えたモデル
  - －  $p < 0.05$  なら「曜日でパラメータに差あり」→ モデルへの組み込みや補正を検討

## Friedman 検定について

Friedman 検定は「同じ被験者が複数条件を繰り返し測定している」場合のノンパラメトリック反復測定 ANOVA

## データの形式

ID が 2 のデータを用いて、計測データの形式を確認した。

File	Dataset	Shape	Dtype
data_2_E4_act.h5	E4	(10, 28799)	float64
data_2_E4_eda.h5	E4	(1, 3599)	float64
data_2_E4_rri.h5	E4	(1, 3600)	float64
data_2_E4_temp.h5	E4	(1, 3599)	float64

例えば、皮膚表面温度データである temp は、4Hz でデータ取得をしている。約 3600 個のデータがあるということは、15 分間のデータである。つまり、このデータは被験者 1 (S01) の 2 回目の感情入力タイミングの直前 15 分の皮膚表面温度の時系列データである。

## データの読み込み方法

以下のようなコードで h5 ファイルの読み込みを行う。

---

Listing 1: HDF5 ファイル読み込み関数

```
import os
import h5py

def list_and_preview_h5_files(directory="sample", preview_shape=(2, 5)):
    """
    指定ディレクトリ内のすべての .h5 ファイルを走査し、
    各データセットの形状と冒頭の一部データを表示します。
    """
    for fname in sorted(os.listdir(directory)):
        if not fname.endswith(".h5"):
            continue
        filepath = os.path.join(directory, fname)
        print(f"\n=== File: {fname} ===")
        with h5py.File(filepath, "r") as f:
            for dset_name, dset in f.items():
                data = dset[()]
                print(f"- Dataset '{dset_name}': shape={data.shape}, dtype={data.dtype}")
    )

if __name__ == "__main__":
    list_and_preview_h5_files()
```

---

## 全データの詳細

以下に、データの形状や値の単位などのメタデータを表形式でまとめる。

ファイル名	説明	データの長さ	単位
data_N_E4_act.h5	x 軸方向の生加速度 (32Hz*15min)	28799	不明
	y 軸方向の生加速度 (32Hz*15min)	28799	不明
	z 軸方向の生加速度 (32Hz*15min)	28799	不明
	身体運動ベクトルの大きさ (32Hz*15min)	28799	不明
	x 軸方向の加速度の重力成分 (32Hz*15min)	28799	不明
	y 軸方向の加速度の重力成分 (32Hz*15min)	28799	不明
	z 軸方向の加速度の重力成分 (32Hz*15min)	28799	不明
	x 軸方向の身体運動成分 (32Hz*15min)	28799	不明
	y 軸方向の身体運動成分 (32Hz*15min)	28799	不明
	z 軸方向の身体運動成分 (32Hz*15min)	28799	不明
data_N_E4_eda.h5	感情計測直前の皮膚電位データ (4Hz*15min)	(1, 3599)	不明
data_N_E4_rri.h5	感情計測直前の心拍間隔データ (4Hz*15min)	(1, 3600)	秒
data_N_E4_temp.h5	感情計測直前の皮膚温度データ (4Hz*15min)	(1, 3599)	摂氏温度

対応するタイムスタンプのデータは、「all-grid-data-extracted2025」から取得できる。計測ファイル名が「data-N-E4-act.h5」であれば対応するタイムスタンプはシートの N 行目を参照すればよい。

## 簡易解析

全被験者のデータをもとに簡単な解析を行う予定だったが、実行環境への全計測データのコピーが丸一日以上かかるため間に合わなかった。次回の ToDo とする。

## 次回の ToDo

- ・ データの傾向を理解するための簡単な解析（一日の計測頻度の分散等）
- ・ どんな回帰による予測が良いか、データの性質を踏まえて考察