

# センサ行動認識におけるモデル選択

大北 剛<sup>1,a)</sup> 井上 創造<sup>1,b)</sup>

概要：三軸加速度センサや角速度センサ（ジャイロセンサ）などのスマートフォン搭載センサや、ドップラセンサや電界共振方式センサなどの非接触式のセンサから取得したデータを基に「歩いている」、「風呂の湯船につかった」などの人間の行動を認識する技術は、IoT/ビッグデータ/人工知能の進展とともに近年急激に注目を集めている。この技術は機械学習を用いて教師ありの学習をする形が一般的 [6], [11] だが、現実的な設定においていわゆる「モデル選択」という話題に到達する。つまり、どのようなパラメータを選択すると推定の精度を最も高く設定するチューニング法に関わる。ウィンドウの幅やサンプリング間隔をどう取ればよいかは自明ではなく、データ依存する場合が多いため、モデル選択を行なう。この技術には副産物があり、どのセンサを用いればよいか、センサ信号のどの部分を重点的に見ればよいかなどの疑問にも答えることができる。本論文の設定においては、ランダムフォレスト、深層学習、アンサンブル RNN などを機械学習器として想定し、これらに対してモデル選択を行なう。モデル選択は従来、パラメトリック型を中心とした AIC[1], BIC[15], MDL[14] などを中心として発展してきたが、ノンパラメトリック型の（深層）学習においてはクロスバリデーションが用いられる。さらに、ハイパーパラメータのモデル選択にはベジアン最適化 [16] やグリッドサーチなどが良く用いられる。また、本論文においては非周期型のモデル選択/特徴選択法であるアンサンブル RNN[10]+ベジアン最適化も導入し、これはベジアン最適化やグリッドサーチなどの周期型のモデル選択とは異なる。アンサンブル RNN をセンサ入力の相対時刻に対して相対的な重要度を特徴選択できるため、従来の方法では周期的に最適なものを選択するしかできなかったが、ここにおいては、周期的に捉われずに時系列で重要な点を直接サンプリングできる形の選択を行なう。これらの上で、グリッドサーチやベジアン最適化 [16] を議論する。

## 1. イントロダクション

三軸加速度センサや角速度センサ（ジャイロセンサ）などのスマートフォンに搭載された携帯デバイスのセンサ技術の進歩から、身に付けることのできるセンサから取得したデータを基に「歩いている」、「ジョギングしている」などの人間の行動を認識する技術が急激に現実のものとなってきた。これと同時に、ドップラセンサや電界共振方式センサなどの非接触式のセンサから取得したデータを基に「風呂に入った」、「風呂で湯船に入った」、「心拍数が 71 である」などの人間の行動を認識する技術も開発されている。これらの技術は IoT/ビッグデータ/人工知能という組み合わせで近年非常に注目され初めている。

加速度センサなどによるセンサ行動認識 [6] の現実的な設定において、どのセンサを用いればよいか、移動窓の幅や移動窓のサンプリング間隔をどう取ればよいか、センサ信号のどの部分を重点的に見ればよいかなどはいずれもシ

ステム全体の性能を決定する重要な情報となる。センサ行動認識の学習器はランダムフォレストなどを用いるが、これは波形の形などの分類をするのみでセンサ行動認識の学習には、移動窓の幅や移動窓のサンプリング間隔なども含んだ形で、センサ行動認識の学習に対してモデル選択する必要がある。この目的のためには、ランダムフォレストなどの学習器のモデル選択より、移動窓の幅や移動窓のサンプリング間隔のパラメータが多いため、本論文においては、グリッドサーチやベジアン最適化 [16] を議論する。グリッドサーチやベジアン最適化は、あらかじめ定義した離散データに対して適用され、この離散データは等間隔であることが多い。一方、アンサンブル RNN を用いたモデル選択においては、センサ入力の相対時刻に対して相対的な重要度に応じた特徴選択を行なえるため、離散データのうち重要度の高い点を中心に直接サンプリングを行なえ、コストを重み付けることができる。したがって、これらの特徴の違いを中心に議論と考察を進める。

## 2. 機械学習としてのモデル選択

統計機械学習においてモデル選択は、予測 (inference) の

<sup>1</sup> 九州工業大学  
〒 804-8580, 北九州市戸畑区山水町 1-1  
<sup>a)</sup> tsuyoshi.okita@gmail.com  
<sup>b)</sup> sozo@mns.kyutech.ac.jp

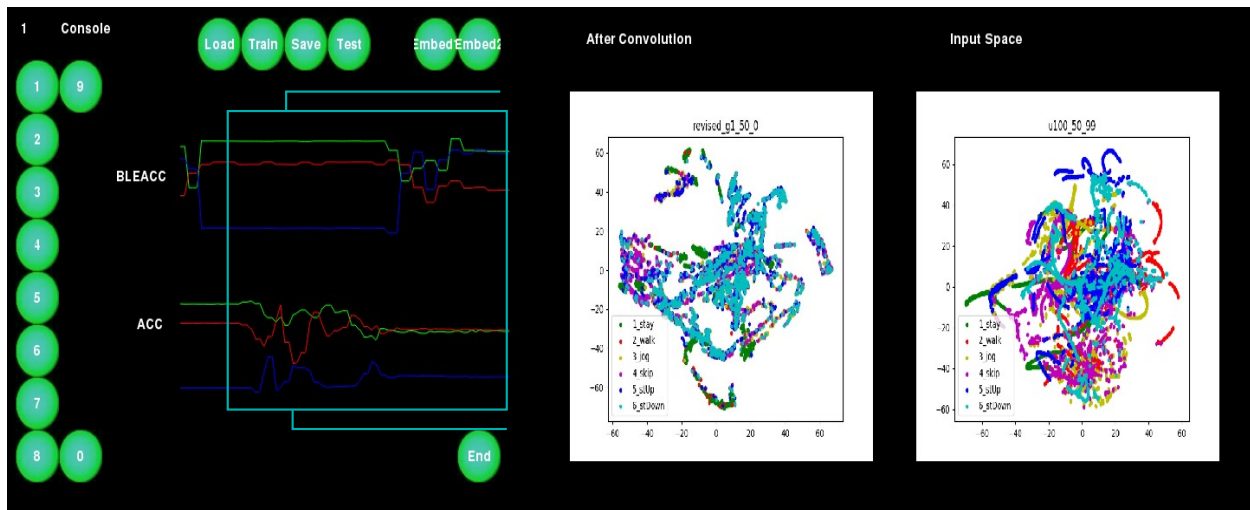


図1 行動認識のコンソールを示す。取得したセンサ信号が6本存在し、青水色の枠が移動窓を示す。1つ目の移動窓と2つ目の移動窓(一部のみが描かれている)の差を移動窓のギャップと呼ぶ。

性能を良くするためにはトレーニングしすぎではないという観点から、トレーニング集合の大きさに対してどの程度の複雑さのモデルが妥当なものであるかを議論する。トレーニング集合の大きさが小さい場合に、複雑すぎるモデルは過学習 (overfitting) のためにテスト誤差が、妥当なモデルより悪くなる。モデルの内部のパラメータを用いて計算する手法には、AIC(赤池情報量基準)[1], BIC[15], MDL[14]などが存在する。一方、クロスバリデーション手法は、評価用のテスト集合を別にとって置き、この評価用のテスト集合に対して性能の良い学習器を選択する手法である。ベイズアン最適化 [16] やグリッドサーチもクロスバリデーションの一種であるがハイパーパラメータの最適化を目的とする。したがって、クロスバリデーションのように5点や10点のサンプリングではなく、グリッドサーチにおいてはパラメータの組み合わせ点数のサンプリングを行なう。組み合わせ点数のサンプリングは計算コストの高いものであるため、ベイズアン最適化においてはガウシアンプロセスなどを用いて、これまでに計算した点の情報を基にして性能の良さそうな点を次に計算するように、計算順序を整えることにより、組み合わせ点数のサンプリングを避ける手法である。最近のアルゴリズムにおいては正則化と捉えられることも多く、ノルムのペナルティーを最適化プロセスに課せる形で機械学習器内に組み込むことも多い [3]。

行動認識におけるモデル選択を扱った論文はあまりないが、ハイパーパラメータの機械学習器への影響をfANOVAフレームワークによって特定する方法によるモデル選択が一件ある [5]。

### 3. ランダムフォレスト/深層学習+ベイズアン最適化によるモデル選択

教師あり行動認識の場合、行動クラス  $\mathcal{Y}$  はトレーニング

集合に有限個の行動として定義され、これはデータによって千差万別となる。たとえば、 $Y = \{ \text{立ち上がる, 歩く, ジョギングする, ...} \}$  などとなる。一方、センサデータ  $S$  は採取に用いるセンサの種類と数に依存し、センサデータの次元は  $n(=\{1, \dots, N\})$  となる。各々のセンサは時系列のデータで構成され、 $k$  番目のセンサ ( $1 \leq k \leq N$ ) に対する時刻を  $t(=\{1, \dots, t_k\})$  と定義すると、 $s^{(k)} = (s_1^k, s_2^k, \dots, s_{t_k}^k)$  と表現できる。但し、サンプリング周波数は必ずしもすべての次元において同じである必要はなく、異なることが多いためにセンサ毎の時刻を考える。以上をより一般的な形で述べるが、その前にいくつかの設定は本論文では議論しない。1つ目に、定義された行動クラスが現実的に十分か否かという議論にも関係するが、ある時刻間のセンサデータは定義された行動クラスに属さないことは現実的にはよくある話だが、この議論は行わない。2つ目にある時刻間のセンサデータが複数の行動の組み合わせ、もしくは行動主が複数であるために行動が複数同時進行することも現実的には起こり得る状況であるが、この議論も除外する。

$N$  有限個の行動クラス  $Y(\subset \mathcal{Y})(= \{y_1, \dots, y_N\})$  を認識する行動認識を考える。  $n(=\{1, \dots, N\})$  次元のセンサを用いて、各々のセンサにおける時刻  $t(=\{1, \dots, t_k\})$  における観測を  $s^{(k)} = (s_1^k, s_2^k, \dots, s_{t_k}^k)$  とする。つまり、観測したセンサデータ  $X$  はサンプリング周期に合えば観測値が存在するが、サンプリング周期に合わない場合には観測値が欠損し、したがって、時刻により欠損する可能性がある。しかし、サンプリング周期に合うような疑似データを得ることはできると仮定すると、 $X(= \{(x_{(1,1)}, \dots, x_{(1,N)}), \dots, (x_{(t_k,1)}, \dots, x_{(t_k,N)})\})$  と書くことができる。一方、慣習にしたがって、固定長の窓の幅を  $w_{width}$ 、考慮する連続する窓の時刻の差 (もしくは、サンプリングする窓の周波数) を

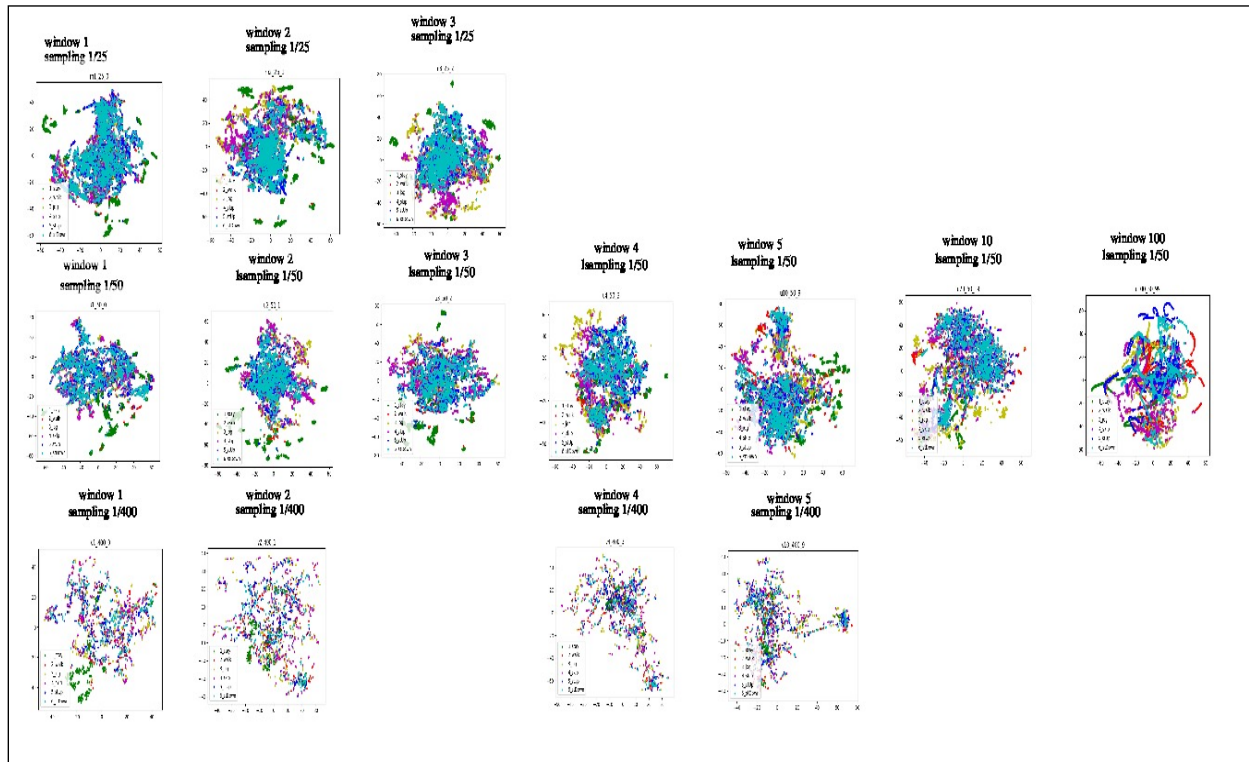


図2 移動窓幅とギャップを t-SNE[18] で描いたもの。データは HASC データを用いた。

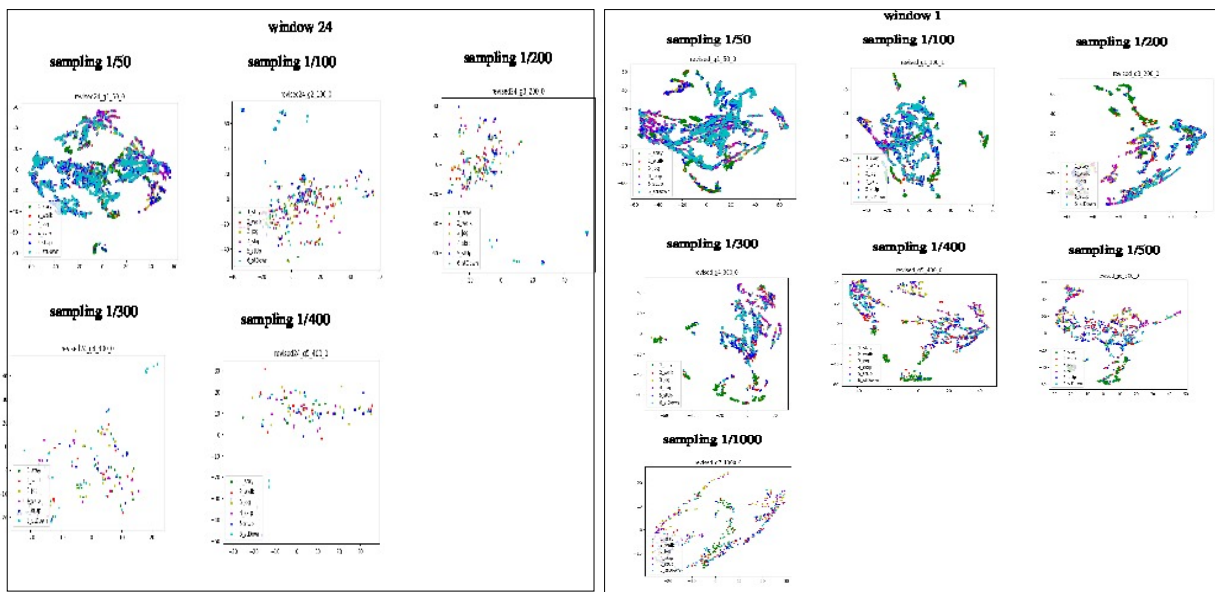


図3 CNN+LSTM モデル [12] での畳み込み層 4 層を通過した後のデータを t-SNE[18] で描いたもの。移動窓幅はこの深層学習モデルの場合には 24(左図) と 1(右図) を選択した。ギャップは図内に描いた値を用いた。データは HASC データを用いた。

$w_{gap}$  とする。これを用いると、これらの  $w_{width}$  と  $w_{gap}$  により、 $X$  は以下の  $X'$  に変換される。  $X' = \{((x_{(1,1)}, \dots, x_{(1,N)}), \dots, (x_{(w_{width},1)}, \dots, x_{(w_{width},N)})), ((x_{(1+w_{gap},1)}, \dots, x_{(1+w_{gap},N)}), \dots, (x_{(1+w_{gap}+w_{width},1)}, \dots, x_{(1+w_{gap}+w_{width},N)})), \dots, (x_{(t-w_{gap}-w_{width},1)}, \dots, x_{(t-w_{gap}-w_{width},N)}), \dots, (x_{(t,1)}, \dots, x_{(t,N)}), \dots, \}$ 。なお、 $z(sensor_t, w_{width}, w_{gap})$  を置くと、 $X' =$

$(z(sensor_1, w_{width}, w_{gap}), \dots, (z(sensor_{t-w_{width}-w_{gap}}, w_{width}, w_{gap})))$  とも表現できる。この  $X'$  を用いて、 $N$  有限個の行動クラス  $Y \subset \mathcal{Y}$  によりラベルづけされたトレーニング集合  $X' \subset \mathcal{X}'$  を用いて、写像  $f: \mathcal{X}' \rightarrow \mathcal{Y}$  を学習する教師あり分類問題を解くことに帰着される。「歩く」という動作の場合、さまざまな歩きかたがあるため、それぞれの歩きかたは異なるセンサ観測を生じる。

移動窓パラメータ  $w_{width}$  と  $w_{gap}$  はセンサ観測データにより大きく変化する。重複の度合いという尺度はこの側面を表現する。 $w_{width}$  が 1 で、 $w_{gap}$  が  $w_{width}$  の場合は、重複度は  $1/w_{width}$  となる。また、 $w_{width}$  が 1 で、 $w_{gap}$  が 1 の場合は、重複度は 1 となる。生のトレーニング集合の大きさを  $u$  とする。重複度を  $o$  とすると、データの大きさは  $o$  倍となる。 $w_{width}$  の大きさは重複度に比例し、一方、 $w_{gap}$  の大きさは重複度に反比例する。行動認識タスクにおいては、これら移動窓パラメータ  $w_{width}$  と  $w_{gap}$  によりトレーニング集合の大きさは劇的に変化する。図 2 はこの劇的さを表現するが、特に 2 段目の右端の図は移動窓の幅を 100 に選択している(実験の項において移動窓の幅が 80 や 100 の場合が特に性能が良かったことを後に述べる)が、連続する移動窓が絵の具をさっと降りかけたようなきれいな模様を描いていることに注目されたい。残念ながら、そのようなきれいな模様は他の移動窓設定では捉えることができなかった。

#### 4. アンサンブル RNN+ベイジアン最適化によるモデル選択

アンサンブルとは RNN を複数個並べて接続したものである。 $m$  番目の RNN モジュールの  $k$  番目の RNN のコンポーネントは以下のように定義できる。

$k$ -th RNN Component

$$\begin{cases} h_{<t>}^{(1)} = \text{LSTM}_{(1)}(h_{<t-1>}^{(1)}, x_t; \tanh) \\ p_k(x_{t,j} = 1 | x_{t-1}, \dots, x_1) = \frac{\exp(w_j^{(1)} h_{<t>}^{(1)})}{\sum_{j'=1}^K \exp(w_{j'}^{(1)} h_{<t>}^{(1)})} \end{cases}$$

各々の RNN は弱学習器ではなく、独立した学習器であるため、エンドツーエンドのアンサンブルとしてのトレーニングは行わず、2 段階でトレーニングを行なう。第 1 段階において、各々の学習器としてディベロップメント集合 1 を用いてトレーニングを行なう。なお、各々の RNN コンポーネントは、さまざまなパラメータのものを用意するものとする。第 2 段階において、各々の RNN をアンサンブルした結果を学習器としての出力とするため、これをディベロップメント集合 2 を用いて、チューニングを行なう。

決定木 [13] において重要度 (variable importance) の概念が導入され、情報量ゲインや Gini インデックスという形の評価尺度が用いられ始めた。情報量ゲインはノードがクラスをいかに不純さなし (impurity) に分割するかを測定し、一方、Gini インデックスはノードがクラスをいかに不平等なし (inequality) に分割するかを測定する。これらを用いた場合に、重要度の高さは必ずしも決定木の上部を意味せず、下部でも重要度の高いものが来る。ランダムフォレストでは、これらの評価尺度は若干不適であったため、Breiman はランダム化した決定木のアンサンブルにおいて、平均減少精度 (mean decrease accuracy) を導入した?。

まず、ブートストラップサンプルとブートストラップ

変数という用語を導入する。ブートストラップサンプル (resample with replacement) を用いて各々の木 (小さな決定木) を学習する部分と、ブートストラップ変数を各分割において用いる部分がこの方式の特徴である。ブートストラップサンプルとはトレーニングデータのランダムな部分集合を指し、したがってこれらの決定木はバイアスされた悪条件 (ill-conditioned) の学習器となり、決定平面は決定木より変化に富んだものとなる。ブートストラップ変数とは、特徴のランダムな部分集合を指し、少量の強い推定器が多数の木で選択されることを防ぐ。

平均減少精度とは以下のような形で定義される。トレーニングデータを、ブートストラップデータ (トレーニングに使うデータ) と OOB(out-of-bag) データ (トレーニングに使わなかったデータ) に分割する。 $i$  番目のランダム変数の重要度を計算するには、まず、OOB データを用いて、 $i$  番目のランダム変数をランダムにシャッフル (置換) した後、決定木に入れて、誤り率  $e_1$  を計算し、次に、シャッフルしない OOB データを用いて、決定木に入れて誤り率  $e_2$  を計算する。これにより、 $e_1 - e_2$  を重要度とする、というのが平均減少精度による重要度の計算である。重要度が小さい場合、変数の値をスクランブルしても木の精度はあまり変化せず、一方、重要度が大きい場合、変数の値をスクランブルすると木の精度は大きく変化する。この尺度をアンサンブル RNN でも用いる [10]。

$$p(x_{t,j} = 1 | x_{t-1}, \dots, x_1) = \sum_{k=1}^M p_k(x_{t,j} = 1 | x_{t-1}, \dots, x_1) \frac{\exp(w_j^{(1)} h_{<t>}^{(1)})}{\sum_{j'=1}^K \exp(w_{j'}^{(1)} h_{<t>}^{(1)})} \quad (1)$$

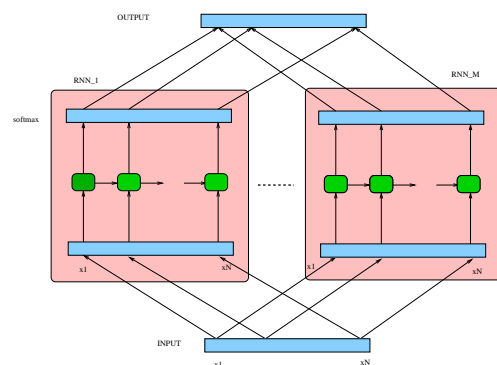


図 4 アンサンブル RNN の概要。入力が下部にあり、出力が上部にある。中間部に図においては RNN 分類器が 2 つ示されている。これらの RNN 分類器の結果を重み付けしたものを出力結果とする。

グリッドサーチ [9] やベイジアン最適化は、時系列の入力のうちの周期的に重要そうな点をサンプルする方法であり、性能が良い周期をサンプルする。一方、アンサンブル RNN

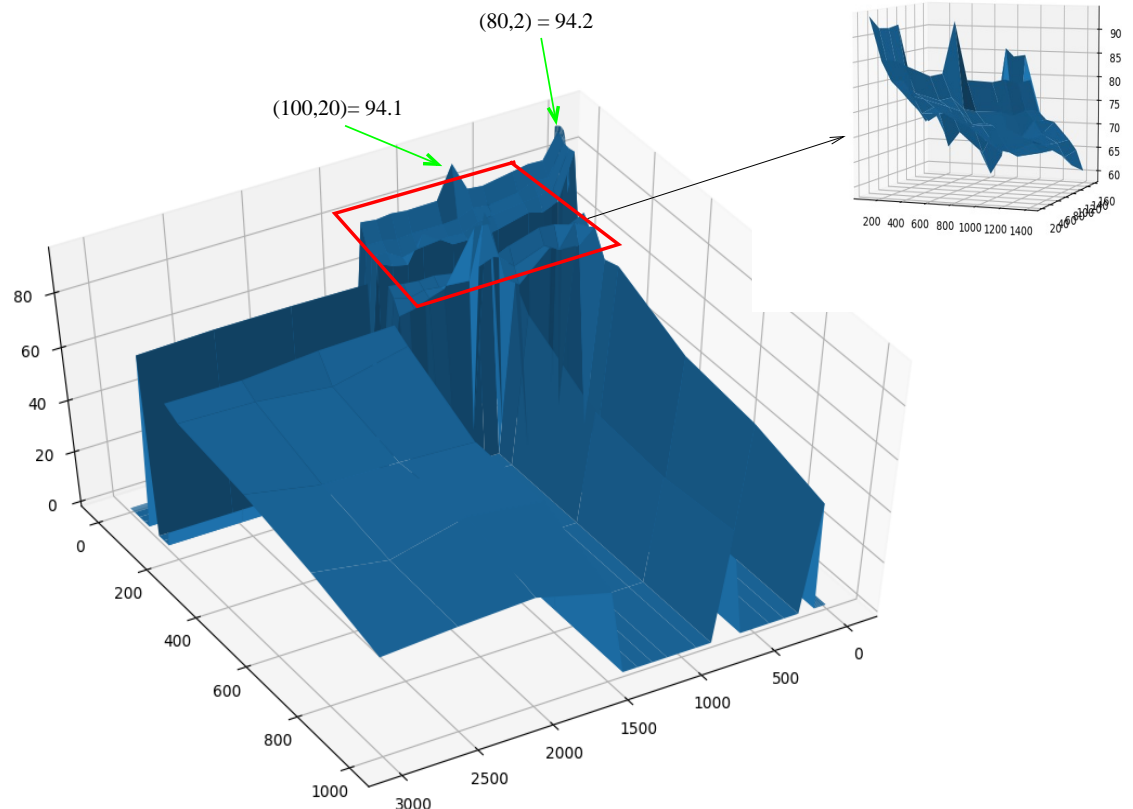


図5 HASCデータの移動窓幅, ギャップに対してモデル選択をした場合のランダムフォレストに対する誤差を示す. 最高の性能は(移動窓, ギャップ)=(80,2)の位置において, 94.2の性能を得た. また, 2番目の性能は(100,20)の地点で, 94.1であった. 図は一部簡略しながら表示している(誤差が0の地点は未評価を示す)が, ピークのすぐ近傍では誤差が大きくなり, ピークの部分だけが鋭い形で突出している. なお, この図はベジアン最適化で得られた図ではない.

はこれより踏み込み, 周期的なことに捉われずに, 特定の分類タスクを行なう際にどの点が特徴として重要かを直接導出できる新たな方法である. ランダムフォレストにおいては, 重要度の高い特徴がどの特徴であるかということを出導できるのと同様, アンサンブルRNNにおいては, 重要度の高い特徴が時系列においてどの特徴であるかということを出導する. なお, 移動窓パラメータ  $w_{width}$  と  $w_{gap}$  に関しては, 時系列なものではないため, アンサンブルRNNでは特に触れない.

## 5. 実験

### 5.1 データ集合

HASCデータセット [7] を実験として用いた. このデータセットの次元は低い, 結果は一般性を失わない形で拡張できると考える. 6行動のラベルを  $\mathcal{Y}$  (=

$\{stay, walk, jog, skip, stairup, stairdown\}$ ) として設定されている. それぞれの行動は10個の時系列のデータで構成されている. 各トレーニング集合はおよそ20秒 ( $t = 1, 856 \sim 2, 028$ ) サンプルされ, サンプル周波数は100Hzで同じものを用いている. 取得したセンサは3次元であり, 加速度センサのX軸, Y軸, Z軸による.

### 5.2 パラメータの設定

移動窓の幅を  $\{500, 1000, 1500, 2000, 2500, 3000\}$  とし, サンプル間隔を  $\{100, 200, 250, 500, 750, 1000\}$  として探索を行なう. このような値域を取ると, たとえば, オリジナルでは60トレーニング集合が1680集合となり, 28倍のサイズの疑似トレーニング集合で学習することとなる. トレーニング集合からランダムに抽出した10%をテスト集合として用いた.

学習器としては以下のものを用いた。ランダムフォレストは scikit-learn のものを用い、CNN+LSTM は Lasagne で実装したもの [11] を用い、アンサンブル RNN は Theano で実装したもの [10] を用いた。

方法	認識率	
ランダムフォレスト/ベイジアン最適化	94.2	
CNN+LSTM/ベイジアン最適化	95.8	
方法	認識率	特徴の数 (割合)
アンサンブル RNN/ベイジアン最適化	82.7	81(33.8%)
アンサンブル RNN/ベイジアン最適化	88.9	89(37.0%)
アンサンブル RNN/ベイジアン最適化	93.8	187(78.0%)
アンサンブル RNN/ベイジアン最適化	93.9	219(91.0%)
アンサンブル RNN/ベイジアン最適化	94.1	227(94.6%)
アンサンブル RNN/ベイジアン最適化	94.0	239(99.6%)

表 1 各最適化の実験結果を示す。HASC データに対して性能が良かったのは CNN+LSTM とベイジアン最適化の組み合わせであった。アンサンブル RNN による非周期的な特徴圧縮はこの実験においてはあまり効果的とはいえなかった。

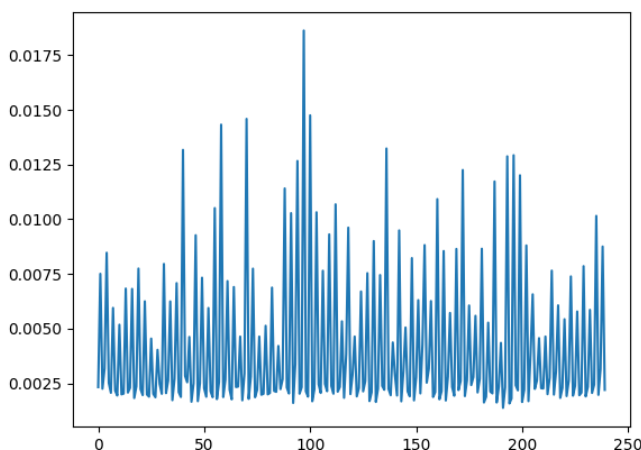


図 6 (移動窓、ギャップ)=(80,2)における時系列に対してアンサンブル RNN を適用し、重要度をプロットした図を示す。横軸は特徴のインデックスを示し、縦軸は重要度の大きさを示す。

## 6. 結論

本論文において、センサ行動認識の問題においてモデル選択を行なった。性能のピーク時は図 3 に示すように突起状になって現われるため、ベイジアン最適化などのモデル選択によりこのピークを的確に捉える必要がある。ランダムフォレスト、CNN+LSTM モデル、アンサンブル RNN を機械学習器として、ベイジアン最適化を行なった。アンサンブル RNN を用いた特徴抽出は、思ったほどの性能には至らず、CNN+LSTM モデルが最も良い性能を挙げた。CNN+LSTM モデルはランダムフォレストより相対値において 2%程度

良い性能であった。なお、HASC データ集合を用いたため、このデータに依存する結果が出た可能性はある。

センサ信号のどの部分を重点的に見ればよいかという問いには、アンサンブル RNN は的確には答えているが、これを用いて抽出した特徴のみではあまり性能が出なかった。78%程度の特徴を用いて 93.8 の性能を得たが、これら以下の特徴量においては急激に悪化することがわかった。

近い将来の展望として、本論文においては 1 種類のデータしか用いていないため、別のデータを用いて比較を行ないたい。時系列性が必要とされるデータにおいてはアンサンブル RNN の有用性が確かめられると予想している。また、センサ次元が高次元であると、機能していないセンサも本手法により発見できるのではと予想している。

## 参考文献

- [1] Akaike, H., "Information theory and an extension of the maximum likelihood principle", in Petrov, B.N.; Cski, F., 2nd International Symposium on Information Theory, : Akademiai Kiado, pp. 267281. 1971.
- [2] L. Breiman, Random forests, Machine Learning, 45:5-32, 2001.
- [3] Cortes, C.; Vapnik, V. "Support-vector networks". Machine Learning. 20 (3): 273297. 1995.
- [4] John Duchi, Elad Hazan, Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization, Journal of Machine Learning Research 12, page 2121-2159, 2011.
- [5] Nils Y. Hammerla, Shane Halloran, Thomas Plotz. Deep, Convolutional, and Recurrent Models for Human Activity Recognition using Wearables. arXiv, 2016.
- [6] 井上創造, ウェアラブルセンサを用いたヒューマンセンシング, 知能と情報, 28:6 pp. 170-186, 日本知能情報フレンジ学会, 2016
- [7] Kawaguchi, N., Ogawa, N., Iwasaki, Y., Kaji, K., Terada, T., Murao, K., Inoue, S., Kawahara, Y., Sumi, Y., and Nishio, N. Hasc challenge: gathering large scale human activity corpus for the real-world activity understandings. In Proceedings of the 2nd Augmented Human International Conference, ACM, 27. 2011.
- [8] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization, the 3rd International Conference for Learning Representations, San Diego, 2015
- [9] Chang, Chih-Chung and Lin, Chih-Jen. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2:3:27:1-27:27, 2011.
- [10] 大北 剛, 井上 創造, シークエンス推定における説明変数の重要度の計算, 情報処理学会自然言語処理研究会研究報告, 5 pages, July, Tokyo.
- [11] Tsuyoshi Okita, Sozo Inoue. Recognition of Multiple Overlapping Activities Using Compositional CNN-LSTM Model. Ubicomp Poster, Sep, 2017.
- [12] Francisco Javier Ordóñez, Daniel Roggen. "Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors 16:115, 2016.
- [13] Quinlan, J. R.. Induction of Decision Trees. Machine Learning 1: 81-106, Kluwer Academic Publishers. 1986.
- [14] Rissanen, J.. "Modeling by shortest data description". Automatica. 14 (5): 465658. 1978.
- [15] Schwarz, Gideon E., "Estimating the dimension of a model",

Annals of Statistics, 6 (2): 461464, 1978.

- [16] Jasper Snoek, Hugo Larochelle and Ryan Prescott Adams. Practical Bayesian Optimization of Machine Learning Algorithms. Neural Information Processing Systems, 2012.
- [17] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions", arXiv e-prints, abs/1605.02688, 2016.
- [18] L.J.P. van der Maaten. Accelerating t-SNE using Tree-Based Algorithms. Journal of Machine Learning Research 15(Oct):3221-3245, 2014.