

身体に装着した加速度センサによる行動およびジェスチャの 統合的認識手法の提案

Joseph Korpela¹ 前川 卓也^{1,5} Julien Eberle² Dipanjan Chakraborty³ Karl Aberer⁴

概要: 本稿では加速度データを用いて行動とジェスチャを同時に扱える統合的な認識フレームワークを提案する。行動認識とジェスチャの認識手法は共に加速度センサを用いているにもかかわらず、行動データとジェスチャデータの特徴は全く異なるため、行動認識とジェスチャ認識手法は全く異なり、これらの研究は別々に行われている。本研究では、それぞれの認識に用いられる特徴や処理手法を用いる弱学習器を複数用意し、それらの弱学習器群を組み合わせることでジェスチャと行動を共に認識可能な決定木を作成する。このとき入力信号の認識に必要と考えられる弱学習器のみを適応的に実行するよう決定木を構築することで、計算コストを抑えた認識を実現する。

1. はじめに

近年、Fitbit や Fuelband, Google watch などの加速度センサを搭載したウェアラブルデバイスが数多く登場している。近い将来、多くの人々がそのようなデバイスを装着し、日々の生活がそれらのデバイス上のセンサデバイスによって継続的に記録され続けるようになるだろう。また、ほぼ全てのスマートフォンは加速度センサを搭載しており、そのセンサから日常生活に関する加速度データを取得し、スマートフォン上で処理することもできる。そのため、このような加速度センサから得られたセンサデータを用いて行動認識を行う研究は、近年の最も活発な研究トピックの1つとなっている。加速度センサにより、例えば「走る」や「スポーツをする」、「歯みがきをする」などの様々な行動が推定できる。

一方、スマートウォッチなどを用いた手によるジェスチャ（ハンドジェスチャ）の認識手法に関する研究も注目を集めている。ハンドジェスチャは、将来のウェアラブルコンピューティング環境のための有力な入力手法の1つであり、スマートウォッチ上の加速度センサを用いて「円を描く」、「手を右に動かす」、などのジェスチャを認識することができる。上述した通り、多くのスマートフォンは加速

度センサを搭載しているが、スマートフォンによるジェスチャ認識も注目を集めている。タッチスクリーンを用いたインタラクションではユーザが画面を注視する必要があるが、ユーザは画面を注視することなくジェスチャによるインタラクションを行うことができる。（例えばボイスメールを再生するコマンドを実行するなど）スマートフォン上の加速度センサを用いて、「スマートフォンをシェイクする」、「スマートフォンを回転させる」などのジェスチャを認識できる。

上述のように、行動認識研究とジェスチャ認識研究は近年注目を集めている研究トピックである。これらの認識手法は共に加速度センサを用いており、ユーザの日常生活から得られるセンサデータは行動に関するデータとジェスチャに関するデータを共に含むことが予想されるにも関わらず、行動認識とジェスチャ認識の研究は別々に行われている。すなわち、行動認識手法は行動のみを認識するように設計されており、ジェスチャ認識手法はジェスチャのみを認識するように設計されている。これは図1に示すように、行動データとジェスチャデータの特徴が全く異なるためである。「走る」や「歩く」などの行動データは時間的に繰り返されるような特徴を持つセンサデータであり、多くの行動認識研究では高速フーリエ変換（FFT）を用いてセンサデータから特徴抽出を行ったあと、抽出された特徴値を連結することにより特徴ベクトルを作成する。そして、特徴ベクトルを機械学習のアプローチ（例えばSVMなど）を用いて、適切な行動クラスに分類する。一方、「円を描く」、「手を右に動かす」などのジェスチャデータは繰り返しのパターンをもたないため、多くのジェスチャ認識研究

¹ 大阪大学
Osaka University
² Ecole Polytechnique Federale de Lausanne
³ IBM Research India
⁴ Ecole Polytechnique Federale de Lausanne
⁵ 国際電気通信基礎技術研究所
Advanced Telecommunications Research Institute
International (ATR)

ではあらかじめ取得したテンプレートと未知のセンサデータを比較し、DTW (dynamic time warping) や隠れマルコフモデル (HMM) などのマッチング手法を用いて、未知のデータが属するジェスチャクラスを特定する。

すなわち、行動情報を利用するアプリケーションと、ジェスチャ情報を利用する他のアプリケーションを同時に稼働しているとき、行動認識を行うプロセスとジェスチャ認識を行うプロセスをスマートフォンもしくはスマートウォッチ上で同時に稼働させる必要がある。これらのプロセスを別々に稼働させることで、加速度データの重複したメモリ確保や重複したセンサデータ処理を引き起こし、無駄な計算コストが発生する。そこで本研究では、行動とジェスチャを同時に扱える統合的な認識フレームワークを設計することを目的とする。

行動とジェスチャを同時に扱えるモデルを実現するために、本研究では行動認識研究やジェスチャ認識研究においてしばしば用いられる特徴抽出や時系列データ処理を基にした弱学習器を複数用意する。例えば、ある軸のセンサデータからエネルギーを計算し、その値を用いてセンサデータセグメントを適切なクラスに分類するような弱学習器を用意する。このような特徴値 (エネルギー) は行動認識にしばしば用いられる。また、例えばジェスチャ認識にしばしば用いられる DTW により、ある軸のセンサデータを適切なクラスに分類するような弱学習器を用意する。さらに、行動認識やジェスチャ認識に共に用いられる特徴 (例えばゼロ交差) を基に認識を行う弱学習器も用意する。このように、行動認識やジェスチャ認識に用いられる特徴 (や手法) を基にした弱学習器を多数用意する。そして、提案手法は入力された未知のセンサデータセグメントを出来るだけ少ない数 (計算コスト) の弱学習器のみを実行して認識することで、低計算コストな行動/ジェスチャの統合認識を目指す。すなわち、不必要な特徴抽出や分析処理が発生しないような手法の実現を目指す。例えば、行動に関するセンサデータが入力される時、行動の認識に必要な弱学習器 (例えばエネルギーやゼロ交差を用いた弱学習器) のみが実行され、ジェスチャの認識に用いられる弱学習器 (例えば DTW など) は実行されない。このように、提案手法は入力されたセンサデータに応じて実行される弱学習器を適応的に切り替える。

このような適応的な弱学習器の選択と実行を実現するため、本研究では木構造のアンサンブル型分類器を提案する。図 2 に例を示す。木構造のそれぞれのノードは弱学習器に対応している。例えば、あるデータセグメントが入力されたとする。まず、根ノードにてデータセグメントの X 軸データから二乗平均平方根 (RMS: the root mean square) が計算される。この計算された RMS 値に応じて次にどの子ノードに進むかが決定される。例えば RMS 値が 3 の場合、「DTW-Y」の子ノードへ進み、さらにデータセグメン

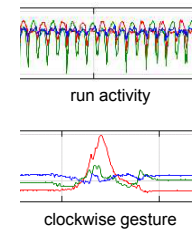


図 1 手首装着型センサデバイスから得た行動とジェスチャセンサデータの例

Fig. 1 Example activity and gesture data from wrist-worn accelerometer.

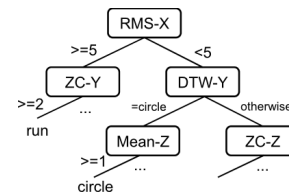


図 2 木構造型認識器の例

Fig. 2 Example tree-structured classifier.

トの分析を続行する。データセグメントを用いた DTW 探索の結果が「円を描く」(circle) であり、さらに Mean-Z の値が 2 であった場合、データセグメントの最終的な分類結果は「円を描く」となる。このような挙動は決定木と類似しているが、提案手法ではそれぞれのノード (弱学習器) においてセグメントのクラス分類確率を計算し、弱学習器の信頼度を考慮してそれらの分類確率を集約し、最終的な分類結果を出力する。また、木はセグメントの認識に要する計算コストを抑えつつ、高い認識精度を達成するよう構築される。

上述の通り、特徴抽出処理 (と DTW) はデータセグメントが通過したノードでのみ実行される。すなわち、データセグメントを正確に認識するために必要なノードのみをセグメントが通過するように木を構築する必要がある。本研究では、ランダムフォレストアルゴリズム [3] を利用して高い認識性能かつ低い計算コストを実現する木を構築することをめざす。

また、本研究では手首装着型デバイス (スマートウォッチなど) もしくはスマートフォン単体による構成および、手首装着型デバイスとスマートフォンのペアからなる構成の 2 パターンの構成を想定し、行動およびジェスチャの認識を行う。手首装着型デバイスもしくはスマートフォン単体による構成では、そのいずれか上で搭載する加速度センサを用いて加速度データを取得し、木を用いてその認識を行う。一方、手首装着型デバイスとスマートフォンのペアからなる構成では、加速度センサを搭載する手首装着型デバイス上で木における処理時間を要さない一部の処理のみを実行したあと、処理時間を必要とする残りの処理はスマートフォン上で行う。このとき、手首装着型デバイスからスマートフォンにセンサデータを送信するために多くの

電力を要する。そこで、手首装着型デバイス上の簡単な処理のみで認識が可能なジェスチャ・行動の場合は、認識結果のみをスマートフォンに送信することで搭載する電池の小さい手首装着型デバイスの省電力化を実現する。本研究では、そのような処理を行う木をトレーニングデータから自動的に学習することを目指す。

本研究の学術的貢献は次のとおりである。(1) 本研究では行動とジェスチャを統合的に認識する手法を、それぞれの認識に用いられる弱学習器を組み合わせることで実現した。(2) 計算コスト削減のため、入力となるセンサデータセグメントの認識に最低限必要な特徴抽出処理（またはDTW処理）のみを実行するようなモデルを設計した。(3) 手首装着型デバイスとスマートフォンのペアからなる構成において、データの送信を削減することで省電力化を実現する手法を設計した。

2. 関連研究

2.1 行動認識

ウェアラブルコンピューティングの研究分野において、身体に装着した加速度センサを用いた行動認識研究がこれまでに多く行われてきた [1], [10], [11]. 多くの研究においては、認識対象とするユーザのラベルありセンサデータから学習したユーザ依存モデルを用いて、行動の認識が行われてきた。近年の研究では他の多数のエンドユーザが取得したラベルありセンサデータを用いる手法が提案されている [2], [13], [14]. これまでの行動認識研究では、エンドユーザがトレーニングデータを収集することを前提としており、本研究でも同様の想定をおく。

計算コストや消費電力を抑えた行動認識を目指す研究もこれまでに多く行われている。これらの研究では、センサを選択的に起動したり、サンプリングレートやデューティサイクルを調整する手法を用いることが多い [4], [19], [20]. これらの技術は本研究と補完的な関係にあるが、本研究では主に行動とジェスチャを共に認識可能な計算コストの低いフレームワークを実現することに焦点を置く。

2.2 ジェスチャ認識

多くのジェスチャ認識研究では、ユーザがジェスチャの開始点と終了点を指定する必要があった [12], [18]. しかし、ウェアラブルコンピュータのような日常的な環境での利用を想定した場合、そのようなユーザによる入力は好ましくない。この問題に対処するため、これまでに連続的な信号を対象としたジェスチャ認識手法に関する研究が行われてきた。しかし、DTWなどの計算コストの高い処理が頻繁に実行される問題があった [8], [21]. この影響を低減するため、ジェスチャが含まれると考えられるデータセグメントを手で設計した軽量なパイプライン処理により検知し、そのときのみDTWなどを実行。一方で、本研究はゼ

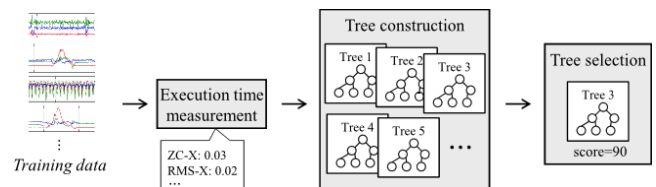


図 3 ラベルありトレーニングデータからの木構造型行動/ジェスチャ認識器構築手法の概要

Fig. 3 Overview of our proposed approach for constructing a tree-structured activity/gesture recognition model from labeled training data.

ロ交差などの軽量な処理を行う弱学習器を用意し、トレーニングデータから自動的に木構造のセンサデータ処理モデルを学習している。

3. 提案手法

3.1 概要

図 3 は木構造型の行動/ジェスチャ認識モデルの構築手法の概要を示す。まず、最初にラベルありトレーニングデータを用いて、それぞれの弱学習器の実行に要する時間を測定する。そしてランダムフォレストアルゴリズムを基に、複数の決定木を作成する。ここでそれぞれの決定木は、行動やジェスチャ認識に用いられる弱学習器群から構成される。最後に、計算コストが小さく高い認識精度が期待される決定木を、作成した決定木群の中から選択し、出力する。

3.2 特徴抽出

本研究では時系列データを扱うため、スライディングウィンドウに含まれるセンサデータセグメントごとに認識を行う。それぞれのノードでは、セグメントから抽出された特徴（またはDTWの結果）を基にどの子ノードに進むかを決定する。本研究では以下に示すような特徴を用意した。

- ゼロ交差 (ZC): セグメント中の原点交差回数を特徴として用いる。ZCはそれぞれの軸ごとに計算する。本研究では3軸加速度センサを想定しているため、それぞれの軸ごとに計3つの弱学習器を用意した (ZC-X, ZC-Y, ZC-Z). 以下の特徴に関しても同様である。
- 二乗平均平方根 (RMS)
- 平均 (Mean)
- 分散 (Var)
- エネルギー (Energy)
- DTW: 本研究ではDTW [17] も弱学習器として木構造の認識器に組み入れる。この弱学習器では、DTW距離k近傍探索を行っており、弱学習器は軸ごとに用意する。既存研究の結果からkは1とし [22], テスト用データセグメントと最も類似するトレーニングセグメントを探索し、結

表 1 特徴抽出の平均計算時間 (Samsung Galaxy S3 を使用)
Table 1 Average computation times for Samsung Galaxy S3

Feature	Time (msec)	Feature	Time (msec)
ZC-X	0.035	Var-X	0.082
ZC-Y	0.028	Var-Y	0.076
ZC-Z	0.028	Var-Z	0.072
RMS-X	0.032	Energy-X	0.780
RMS-Y	0.027	Energy-Y	0.787
RMS-Z	0.027	Energy-Z	1.346
Mean-X	0.067	DTW-X	376.557
Mean-Y	0.043	DTW-Y	365.114
Mean-Z	0.061	DTW-Z	371.219

果を出力する。上述の ZC や RMS のような数値を出力する弱学習器と異なり、この弱学習器はノミナルの情報を出力する。(例えば「歩く」や「円を描く」といった行動やジェスチャのクラスの情報)

3.2.1 計算コスト

上述したように、さまざまな弱学習器を用意しており、それぞれの計算コストをあらかじめ計算しておく。表 1 は Samsung Galaxy S3 を用いて 1 秒の窓幅のデータセグメントを処理した際の、それぞれの弱学習器の計算時間である。この結果から、DTW の計算時間が他の弱学習器に比べて大きいことが分かる。

3.3 木の学習と認識

まず、手首装着型デバイスもしくはスマートフォン単体による構成における、木構造型認識器の学習とテストデータの認識について述べる。

3.3.1 木ノード

提案する木構造には 2 種類のノード (弱学習器) が含まれる。1 つ目は、ZC や RMS などの数値の特徴を用いて決定を行うノードであり、2 つ目はノミナルの特徴 (DTW の結果) を用いて決定を行うノードである。それぞれのノードでは、センサデータセグメントから抽出された特徴の値に基づいてどの子ノードに進むかを決定する。

[数値の特徴を用いるノード]

数値の特徴を用いるノードでは 2 値分類 (2 分岐) を行う。図 2 の例を用いて説明すると、RMS-X の値が 5 より大きい時、テストセグメントは ZC-Y の子ノードに進む。それ以外の場合は DTW-Y の子ノードに進む。この閾値 (分割) は、C4.5 決定木 [15] と同様にトレーニングデータの情報利得比が最大となるように選択する。

[ノミナルの特徴を用いるノード]

ノミナルの特徴を用いるノードにおいても、情報利得比が最大となるような分割を用いる。このとき、ノミナルを用いた場合の分割には主に以下の 3 つの手法が用いられており、評価実験ではそれぞれを検証する。

- *Binary*: ノードにおいて 2 値分類 (2 分岐) を行う。
- *Multway*: それぞれのノミナル値ごとに子ノードへと進む枝を用意する。
- *Multi-subset*: ノミナル値のサブセットごとに子ノードへと進む枝を用意する。例えば、「走る」または「歯磨きをする」のノミナル値に対応する際に進むノードを用意する。情報利得比が最大となるようなサブセットを用いた分割は、k-means アルゴリズムとエン트로ピーを用いたクラスタリング手法を基に決定する [7], [9]。

3.3.2 木の構築と選択

まず、複数の決定木をランダムフォレストアルゴリズムを基に構築する。ランダムフォレストアルゴリズムでは、C4.5 アルゴリズムを用いて木を構築している。本研究では、あるノードの分割を決定するとき、ランダムフォレストアルゴリズムを基にランダムに選んだ特徴セット (弱学習器) のみから、最適な分割を決定する。ただしランダムフォレストアルゴリズムとは異なり、ランダムに特徴 (弱学習器) セットを選択する際に、弱学習器の計算コストに応じた重みを付けて選択する。 i 番目の弱学習器が選択される確率は $Pr(i) = 1 - c \frac{cost_i}{\max_{n \in W} (cost_n)}$ のように表される。ここで、 $cost_i$ は i 番目の弱学習器の計算コストであり、 W は弱学習器のセット、 c は計算コストの重みをコントロールする定数である。このように確率に偏りを持たせることにより、計算コストの大きい弱学習器はノードとして用いられづらくなり、木構造全体としての計算コストが下がる。このようにランダムに特徴 (弱学習器) のセットを取得したあと、情報利得比を最大化する特徴およびその分割を決定する。

上記の方法を繰り返して複数の木を生成したあと、許容できる計算コスト内に収まり、高い認識精度が期待されるような木を選択する。木の計算コストは、トレーニングデータを用いて実際に計算を行うことで求めることができる。木の認識精度はテストデータを用いて求めるべきであるが、事前にテストデータを入手することは不可能であるため、本研究では木に含まれるノードの数に着目して木の認識精度を推定する。一般的に、ノード数の多い木はトレーニングデータにオーバーフィットしていると考えられ、ノードの数が少ない木ほどの高い木であると言える [16]。事前にランダムに作成した 210 の木を検証したところ、木のサイズ (ノード数) と認識精度 (F 値) には -0.861 の逆相関が見られた。そこで本研究では、指定した許容できる計算コスト内に収まり、ノード数が最も小さい木を、最終的な認識モデルとして出力する。

3.3.3 テストデータの認識

上述のようにして求められた木を用いて、テストデータセグメントを認識する。このとき、テストデータセグメントが通過するノードのみでしか特徴抽出処理が行われない。ここで、セグメントが DTW を用いた弱学習器を通過しな

い場合、その挙動は通常の決定木と同様となる。一方で、DTW を用いた弱学習器を通過する場合、DTW の計算コストが他の弱学習器に比べて膨大な計算コストを要するにも関わらず、DTW の結果が最終的なセグメントの分類結果に大きく反映されているとは言い難い。図 2 の例を用いて説明する。セグメントがルートノードから DTW-Y ノードに進んだとし、DTW-Y の結果が「円を描く」であるとする。この DTW の結果は DTW-Y のどの子ノードに進むかを決定するのみに用いられる。すなわち、DTW-Y の結果は Mean-Z における決定において直接的に用いられるわけではない。提案手法では、テストセグメントが DTW を用いたノードを通過した場合はノードにおいて分類されるクラスの再計算を行う。具体的には、セグメントが通過したノード（弱学習器）が出力したクラス分類確率を基に投票により再集計する。

投票は各弱位置推定器により行われ、票の重みは弱学習器の確信度によって決まる。あるテストセグメント f_t が n 番目のクラス (C_n) に最終的に分類される確率は以下の式で表わされる。

$$Pr(f_t|C_n) = \frac{1}{\sum_{m=1}^M w_m} \sum_{m=1}^M w_m Pr_m(f_t|C_n), \quad (1)$$

ここで、 M は f_t が通過した弱学習器の数、 w_m は m 番目の弱学習器の票の重み、 $Pr_m(f_t|C_n)$ は m 番目の弱学習器により出力された f_t が C_n に分類される確率である。重みは、弱学習器の確信度に対応する。DTW を用いた弱学習器においては、トレーニングデータから求められた F 値を重みとして用いる。それ以外の数値の特徴を用いる弱学習器においては、トレーニングデータから求められた情報利得を用いる。図 2 の例を用いて説明する。100 のトレーニングセグメントが RMS-X ノードから ZC-Y ノードへ進んだとする。この 100 のセグメントは、30 の「走る」と 70 の「歯を磨く」からなるとする。これらのセグメントの最終的なクラス分類確率を計算するとき、RMS-X ノードの重みはこれら 100 のセグメントのエントロピーを用いて計算される。エントロピーはインスタンスがクラス分類される際の不確かさを表し、以下の式で記述される。

$$w_m = 1 - \frac{H(P)}{\log(C,2)}$$

$$H(P) = -\sum_{n=1}^C p(\mathbf{f}_m, C_n) \log p(\mathbf{f}_m, C_n),$$

ここで、 C は行動/ジェスチャクラス数、 \mathbf{f}_m は注目する枝（この例では RMS-X の左の枝）を通過したトレーニングセグメント（インスタンス）群、 $p(\mathbf{f}_m, C_n)$ は \mathbf{f}_m に含まれる C_n に属するセグメントの割合を示す。例えば、「走る」クラスの場合、 $p(\mathbf{f}_m, C_n) = \frac{30}{100}$ となる。

式 1 において DTW を用いたノードの $Pr_m(f_t|C_n)$ を計算するとき、DTW 探索の結果が C_n の場合は $Pr_m(f_t|C_n) = 1$

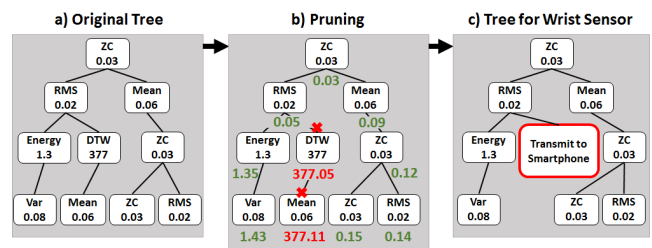


図 4 手首装着型センサデバイスを用いた場合の木構造型行動/ジェスチャ認識器構築手法の概要

Fig. 4 Overview of modification made to tree-structured activity/gesture recognition model for use in a wrist-worn sensor.

とし、それ以外は $Pr_m(f_t|C_n) = 0$ とする。その他のノードに関しては、 $Pr_m(f_t|C_n) = p(\mathbf{f}_m, C_n)$ とする。

以上のように、テストセグメントが通過した弱学習器の確信度を考慮して最終的な分類を行う。

3.4 手首装着型デバイス/スマートフォンのペアのための木の構築と認識

手首装着型デバイスとスマートフォンのペアからなる構成における認識について述べる。基本的に、上記で作成した決定木を改良して用いる。

3.4.1 木構造型認識器の改良

上述したように、手首装着型デバイス上で認識可能な行動/ジェスチャの認識を行ってしまうことで、スマートフォンに転送するセンサデータの量を削減する。これを実現するため、木の作成アルゴリズムおよびテストデータの認識方法を改良する。手首装着型デバイスは計算能力が貧弱であるため、DTW のような計算コスト（電力消費）を要求する処理を行うことは難しい。そこで、手首装着型デバイス上では指定された閾値（消費電力）以下の処理のみを行うように木の枝刈りを行う。そして、枝刈りをした枝以下の部分木に対応する処理はスマートフォン上で行う。部分木のコストは、テストデータセグメントが通る木のパスによって異なるため、本研究では最も計算コストが小さくなるパスのコストを部分木のコストとする。これにより、手首装着型デバイス上で計算コストの高いノードの処理が実行されることを防ぐ。ここで、あるパスにおけるコストは、そのパス上の全てのノードのコスト（消費電力）の合計とする。すなわち、部分木のコストは、その部分木の根ノードからのパスのうち、最もコストの小さいパスのコストに対応する。

本研究では木を根ノードから葉ノードまで辿って、コストが閾値より大きい部分木を線形時間で発見する。具体的には、あるノードを辿る際、そのノードに対して、そのノードの弱学習器の計算コストとその親ノードに割り当てられたコストの和を割り当てる。そのノードに割り当てられたコ

ストの和は、そのノードに到達したときに手首装着型センサデバイス上で実行される処理のコストを示す。その後さらに根ノードから辿って、あらかじめ設定した閾値より初めてコストが大きくなるノードを根とする部分木を全て枝刈りする。図 4 示す通り、枝刈りされた部分木は、センサデータをスマートフォンに送信することを示すノードに置き換えられる。手首装着型デバイス上では部分木が削除されたの木に対応する処理のみが行われ、スマートフォンでは、部分木に対応する処理が手首装着型デバイスの代わりに行われる。また、手首装着型デバイス上の処理葉ノードまで到達した場合は、その認識結果のみをスマートフォンに送信し、センサデータはスマートフォンには送信しない。これにより大幅に通信に関わる電力消費を削減することができる。

3.4.2 手首装着型デバイス/スマートフォンのペアによる認識

上述した通り、サイズの小さな手首装着型デバイスはサイズの小さなバッテリーしか搭載できないため、省電力な認識が必要である。特に Bluetooth によるデータ送信の消費電力は大きく、Bluetooth Low Energy (BLE) を用いたとしても、データの送受信は小型デバイスのバッテリー稼働時間を 85%も短縮させてしまうと言われている [6]。上述の木の枝刈りを行う際の閾値を適切に設定し、手首装着型デバイス上で認識可能な行動/ジェスチャに関するセンサデータの送信を行わないことで、データ送信に関わる電力消費を大幅に削減できる。具体的には、閾値をデータの送信に必要な消費電力と同じ値に設定することで、手首装着型デバイス上ではデータの送信以上に電力を消費する処理を行わず、代わりにスマートフォン上で実行させることができる。手首装着型デバイス上で認識が終了した場合は、その認識結果のみをスマートフォンに送信する。認識結果はそのデータ量が非常に小さく、行動/ジェスチャクラスの数を C とすると、 $\log(C)$ ビットのオーダーとなる。例えば行動/ジェスチャクラスの数が 256 の場合、1 バイトのオーダーで表現できる。

3.4.3 センサデータ送信における送信データ量削減の効果

BLE プロトコルをセンサデータの送信に用いた場合、パケットごとに最大で 20 バイトのペイロードデータ（データの本体）に加えて 19 バイトのオーバーヘッドが送信される [5], [6]。1 軸の加速度データの 1 サンプルが 10 ビットで表現される場合、3 軸のデータを合わせて 30 ビットのペイロードデータが必要となる。1 秒の窓幅で 50%オーバーラップのスライディングウィンドウを用いたとき、100Hz のセンサデータの場合、1 つのウィンドウのデータを送信するために 188 バイトのペイロードデータが送信される。(50%オーバーラップしているため、ウィンドウ内の半分のデータを送ればよい。) このデータは 10 個の BLE パケットに分けて送信する必要があるため、オーバーヘッドを含

めて 378 バイトのデータが送信される。一方、手首装着型デバイス上で行動/ジェスチャの認識が完了した場合はその結果 (1 バイト) のみを送信するため、オーバーヘッドを含めて 20 バイトのみが送信される。これにより、94.7%のデータ量の削減が実現できる。

また、本研究では 50%オーバーラップのスライディングウィンドウを用いているため、手首装着型デバイスが 1 秒幅のウィンドウのデータを全て送信する必要がない。これは、ウィンドウの前半のデータが前時刻のウィンドウ内のデータ送信の際に送信済みのためである。しかし、前の時刻においてデータ送信が行われなかった場合、手首装着型デバイスは前半のデータも送信する必要がある。すなわち、10 個の BLE パケットを追加で送信する必要があるため、送信時間が 2 倍になり、認識結果の出力に遅延が生じてしまう。しかし、1 個のパケットを送信するのに要する時間は 0.72 ミリ秒程度言われており、10 個のパケットを送信したとしても遅延は追加で 7.2 ミリ秒となり、その遅延は非常に小さい。

3.4.4 認識結果送信回数の削減

多くの行動は、ある程度長時間連続して起こる。例えば、歩く行動などは数十秒続くことが多いだろう。手首装着型デバイスからこのような連続した結果をまとめて送ることで、通信パケットを削減することができるが、認識結果をリアルタイムにスマートフォンに送信できなくなる問題が生じる。そこで本研究では、認識結果を利用するアプリケーションによって、結果を必要とする間隔が異なることに着目する。例えば、ジェスチャ入力コマンドを扱うようなアプリケーションは、ジェスチャの認識結果をリアルタイムに必要とするだろう。一方、行動認識結果を記録するようなライフログなどのアプリケーションは、リアルタイム性を必要としない。すなわち、リアルタイムに結果を送信する必要のない認識結果をバッファに格納し、適切なタイミングでまとめて送信することで送信に関する消費電力を削減する。最大で 20 の認識結果 (1 つの結果は 1 バイト) をまとめて送ることで、その都度送る方法に比べて 400 バイトから 39 バイトまで送信データ量を削減できる。

4. 評価実験

4.1 データセット

今回は、手首装着型デバイス単体からなる構成を想定した評価実験のみ行う。本研究では、100Hz のサンプリングレートをもつ 3 軸加速度を搭載した無線センサノードを用いてデータを収集した。センサノードは 3 人の参加者の右手首に装着してもらった。実験では、それぞれの参加者が表 2 に示した行動とジェスチャをランダムな順でひと通り行うセッションを 10 回ずつ行なった。表に示す行動とジェスチャは、既存の研究で用いられているものを参考に選んだ。

表 2 実験で用いた行動/ジェスチャのリスト。(カッコ内はデータセットに含まれるインスタンス数の比率)

Table 2 Activities/gestures performed in our experiment associated with proportions of instances included in sensor data.

	activity		gesture
A	run (18%)	F	left to right (1%)
B	draw on whiteboard (19%)	G	right to left (1%)
C	wash dishes (18%)	H	clockwise (1%)
D	write in notebook (19%)	I	counter-clockwise (1%)
E	brush teeth (18%)	J	down to up (1%)

4.2 評価方法

本研究では、「leave-one-session-out」交差検定を用いて評価を行った。すなわち、1つのセッションをテストデータ、残り9つのセッションをトレーニングデータとして交差検定を行う。提案手法の有効性の評価のため、以下の3つの手法を提案手法に加えて用意した。

- *Act*: この手法では、行動認識に一般的に用いられる特徴のみを用いて決定木で認識を行う。すなわち、DTW は用いない。決定木には Weka データマイニングツールキットに実装されている C4.5 (J48) 決定木をデフォルトの設定で用いた。

- *DTW*: この手法は DTW を基にした k 近傍探索により認識を行う。このとき、それぞれの軸のデータに対して k 近傍探索を行い、多数決で最終的な認識結果を決定する。

- *Tree*: DTW の結果と数値の特徴を共に用いる C4.5 決定木を構築し、認識に用いる。この手法では、テストデータセグメントが通過したノードのみで特徴計算処理が行われる。

- *Proposed*: 本提案手法である。

これらの手法の評価には、平均 F 値を用いる。また、計算コストとしては、Galaxy S3 スマートフォンで1つのテストデータセグメントを処理するのにかかる平均時間を用いる。

4.3 結果

4.3.1 ベースライン手法の認識精度

表 3 は、*Act*, *DTW*, *Tree* の平均 F 値と計算コストを示す。結果に示す通り、*Act* は計算コストを大幅に抑えているが、その認識精度は低かった。これは、計算コストが高く、ジェスチャの認識に貢献する DTW を用いていないためと考える。一方で、*DTW* の F 値は *Act* に比べて向上しているが、計算コストは非常に高い。また、*DTW* の F 値は *Tree* に比べてそれほど高くなかった。特に、「draw on whiteboard」と「clockwise」の F 値が低かった。*Tree* の平均 F 値は高かったが、計算コストも同様に高かった。これは C4.5 アルゴリズムが情報利得比の大きい弱学習器を根に近いノードに用いるように木を構築するため、DTW

表 3 *Act*, *DTW* および *Tree* の計算コストと平均 F 値

Table 3 The computation cost and average F-measure for *Act*, *DTW* and *Tree*.

	cost (msec)	avg. F-measure
<i>Act</i>	0.38	0.892
<i>DTW</i>	1112.89	0.932
<i>Tree</i>	388.14	0.962

を用いた弱学習器が根に近いノードに配置されたからと考える。実際に、*Tree* により作成された全ての木において、DTW を用いた弱学習器が根となっていた。これにより、全てのテストセグメントを認識する際に DTW が実行されるため、計算コストが高くなる。以降では提案手法である *Proposed* と *Tree* の計算コストおよび平均 F 値を比較する。

4.3.2 提案手法の結果

表 4 に提案手法の結果を示す。DTW を用いた弱学習器の分割方法は 3.3.1 節に示す通り 3 種類用意しており、それぞれごとに *Tree* と比較した計算コストと精度を示している。例えば、コストが 15.1% の場合、*Tree* と比べて計算コストが 15.1% に抑えられていることを示す。また、表 4 には閾値を変化させたときの結果も示している。ここで閾値とは許容可能な計算コスト(ミリ秒)を示し、提案手法は閾値よりコストが小さく、最もサイズの小さい木を出力する。この閾値により木の計算コストをコントロール可能である。表 4 の上部 3 つの結果が、コストを約 15% 程度に低減するよう閾値を設定した例である。また、「accuracy(%) w/o voting」は弱学習器の分類結果を集約しない場合の精度比を示す。ここで精度比は *Tree* の平均 F 値と比較した精度である。これら 3 つの結果では、コストを 15% 程度にまで削減しているにも関わらず、96 から 97% 程度の精度比を達成している。また表 4 には、木に含まれるノードの平均数も示している。*Multitway* の分割方法を用いた場合、木に含まれるノード数が最も小さい。これは、1つのノードの分岐数が多いため、少数のノードのみでも分類性能の高い木を構築できたからと考える。

さらに表 4 の「accuracy(%) w/ voting」には、弱学習器の結果を集約した場合の結果も示している。結果の集約を行うことでさらに精度は向上し、98 から 100% の精度を達成している。

表 4 には、閾値を大幅に小さく設定した (25 ミリ秒) 結果も示している。*Multitway* は計算コストを 2.7% にまで削減したうえ、99.8% の精度比を達成している。2.7% の計算コストは、10.5 ミリ秒の計算時間に対応し、大幅に計算コストを削減できているにも関わらず、高い認識精度を達成した。上述した通り、*Multitway* はそれぞれの行動/ジェスチャクラスごとに DTW ノードの枝を構築するため、低い計算コストを達成したと考える。また、結果を集

表 4 Tree に対する Binary, Multiway および Multi-subset の計算コストと平均 F 値の比率 (%)

Table 4 Costs and accuracies (%) for Binary, Multiway, and Multi-subset when compared with Tree.

	cost (%)	accuracy(%) w/o voting	accuracy(%) w/ voting	avg. # of DTW nodes
Binary TH = 100	15.1%	96.6%	99.2%	4.6
Multiway TH = 250	13.4%	97.9%	98.7%	1.7
Multi-subset TH = 100	17.6%	96.7%	100.2%	3.2
Binary TH = 25	2.8%	95.9%	96.7%	2.9
Multiway TH = 25	2.7%	97.0%	99.8%	1.5
Multi-subset TH = 25	2.0%	95.8%	97.3%	1.5

約することで約 5.3%の精度向上を実現している (88.9%から 94.2%)。さらに分析の結果、行動に対応するセグメントのうち 1.5%のみが DTW を用いたノードを通過していた。すなわち、提案手法により行動の認識に大きく貢献しない DTW の処理を自動的に削減できていた。一方、ジェスチャに対応するセグメントのうち 59.7%が DTW を用いたノードを通過していた。この結果から、いくつかのジェスチャクラスの認識には DTW を用いずとも高い精度を達成できるものと考えられる。提案手法ではこれを自動的にトレーニングデータから学習し、認識モデルを構築していた。

5. おわりに

本研究では行動とジェスチャを統合的に認識可能な新しい手法の提案をおこなった。提案手法では、それぞれの認識に用いられる特徴や処理手法を基にした弱学習器を複数用意し、入力信号に応じて適応的に弱学習器を実行することで、計算コストを抑えた認識を実現した。また、評価実験により提案手法の有効性を確認した。

参考文献

[1] Bao, L. and Intille, S.: Activity recognition from user-annotated acceleration data, *Pervasive 2004*, pp. 1–17 (2004).

[2] Berchtold, M., Budde, M., Gordon, D., Schmidtke, H. and Beigl, M.: ActiServ: Activity recognition service for mobile phones, *International Symposium on Wearable Computers*, pp. 1–8 (2010).

[3] Breiman, L.: Random forests, *Machine learning*, Vol. 45, No. 1, pp. 5–32 (2001).

[4] French, B., Siewiorek, D. P., Smailagic, A. and Deisher, M.: Selective sampling strategies to conserve power in context aware devices, *Wearable Computers, 2007 11th IEEE International Symposium on*, IEEE, pp. 77–80 (2007).

[5] Gomez, C., Demirkol, I. and Paradells, J.: Modeling the maximum throughput of Bluetooth low energy in an error-prone link, *Communications Letters, IEEE*,

Vol. 15, No. 11, pp. 1187–1189 (2011).

[6] Gomez, C., Oller, J. and Paradells, J.: Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology, *Sensors*, Vol. 12, No. 9, pp. 11734–11753 (2012).

[7] Hartigan, J. A. and Wong, M. A.: Algorithm AS 136: A k-means clustering algorithm, *Applied statistics*, pp. 100–108 (1979).

[8] Junker, H., Amft, O., Lukowicz, P. and Tröster, G.: Gesture spotting with body-worn inertial sensors to detect user activities, *Pattern Recognition*, Vol. 41, No. 6, pp. 2010–2024 (2008).

[9] Li, X., Zhang, K. and Jiang, T.: Minimum entropy clustering and applications to gene expression analysis, *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pp. 142–151 (2004).

[10] Lukowicz, P., Junker, H., Stager, M., Buren, T. V. and Tröster, G.: WearNET: A distributed multi-sensor system for context aware wearables, *UbiComp 2002*, pp. 361–370 (2002).

[11] Lukowicz, P., Ward, J., Junker, H., Stager, M., Tröster, G., Atrash, A. and Starner, T.: Recognizing workshop activity using body worn microphones and accelerometers, *Pervasive 2004*, pp. 18–32 (2004).

[12] Lyons, K., Brashear, H., Westeyn, T., Kim, J. S. and Starner, T.: Gart: The gesture and activity recognition toolkit, *Human-Computer Interaction. HCI Intelligent Multimodal Interaction Environments*, Springer, pp. 718–727 (2007).

[13] Maekawa, T. and Watanabe, S.: Unsupervised Activity Recognition with User’s Physical Characteristics Data, *Int’l Symp. on Wearable Computers*, pp. 89–96 (2011).

[14] Ohmura, R., Hashida, N. and Imai, M.: Preliminary evaluation of personal adaptation techniques in accelerometer-based activity recognition, *Int’l Symp. on Wearable Computers: Late Breaking Results* (2009).

[15] Quinlan, J. R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann (1993).

[16] Quinlan, J. R.: Improved use of continuous attributes in C4. 5, *arXiv preprint cs/9603103* (1996).

[17] Sakoe, H. and Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, Vol. 26, No. 1, pp. 43–49 (1978).

[18] Schlömer, T., Poppinga, B., Henze, N. and Boll, S.: Gesture recognition with a Wii controller, *Proceedings of the 2nd international conference on Tangible and embedded interaction*, ACM, pp. 11–14 (2008).

[19] Shih, E. I., Shoeb, A. H. and Guttag, J. V.: Sensor selection for energy-efficient ambulatory medical monitoring, *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ACM, pp. 347–358 (2009).

[20] Wang, Y., Lin, J., Annavaram, M., Jacobson, Q., Hong, J., Krishnamachari, B. and Sadeh, N.: A framework of energy efficient mobile sensing for automatic user state recognition, *MobiSys 2009*, pp. 179–192 (2009).

[21] Wu, J., Pan, G., Zhang, D., Qi, G. and Li, S.: Gesture recognition with a 3-d accelerometer, *Ubiquitous intelligence and computing*, Springer, pp. 25–38 (2009).

[22] Xi, X., Keogh, E., Shelton, C., Wei, L. and Ratanamahatana, C. A.: Fast time series classification using numerosity reduction, *Proceedings of the 23rd international conference on Machine learning*, ACM, pp. 1033–1040 (2006).