

共通パターンを用いた時系列データの弱教師ありセグメンテーション

入船 恭彰[†] 松原 靖子^{††} 佐藤 貴大[†] 有次 正義^{†††} 櫻井 保志^{††}

[†] 熊本大学大学院 自然科学教育部 〒860-8555 熊本県熊本市中央区黒髪2丁目39番1号

^{††} 大阪大学 産業科学研究所 〒567-0047 大阪府茨木市美穂ヶ丘8-1

^{†††} 熊本大学大学院 先端科学研究部 〒860-8555 熊本県熊本市中央区黒髪2丁目39番1号

E-mail: [†] {irifune,stakahiro}@st.cs.kumamoto-u.ac.jp, ^{††} {yasuko,yasushi}@sanken.osaka-u.ac.jp

^{†††} aritsugi@cs.kumamoto-u.ac.jp

あらまし 近年、深層学習を用いたモデルが多く分類問題において高精度を達成している一方、モデル開発におけるデータ準備の重要度が高くなっている。一般的な分類問題では所望の出力と同様のデータとラベルが必要であり、人手で行われることが多いラベル付け作業には大量の時間とコストを消費する。センサーやログデータといった時系列データにおいても同様であるが、画像とは異なりデータを観察しアノテーションを行うことは困難である。例えば、センサーデータを用いた行動認識では、(時刻, ラベル)の2つ組で構成される「いつ, 何をしていたか」という情報を観察者あるいは撮影した動画を利用して記録しなければならない。そこで本論文では、複数のパターンを含む時系列データのための弱教師を用いたセグメンテーション手法である LABELGEAR を提案する。LABELGEAR は各シーケンスに「何をしていたか」を表すラベルのみが1つ以上付与されたデータに対し、各時刻におけるラベルを分類することによりセグメンテーションを行う。実データを用いた実験では、提案手法がウェアラブルセンサーデータから各ラベルの特徴を抽出し、高精度にラベルを分類していることが確認でき、アノテーションの効率化を示した。

キーワード 時系列データ, 深層学習, 弱教師あり学習, セグメンテーション

1 はじめに

近年、深層学習がめざましく発展し、あらゆる分野において高い分類精度を達成するモデルが数多く報告されている[2],[12],[14],[19]。深層学習は End-to-End 学習と呼ばれ、データとラベルから入出力関係を直接学習することにより、従来の人手により設計された特徴量による学習に比べて高い精度を実現する。これは、モデル開発において学習データの量と質がモデルの性能を左右することを意味し、いかに効率よく高品質なラベル付けを行い、大量のデータを揃えるかが重要な課題となる。

一般的に、分類問題では所望の出力と同様のデータとラベルを求められる。アノテーションと呼ばれるラベル付け作業は人手で行われることが多いため、大量の時間とコストを消費することが問題となる。特に、センサーデータやログデータといった時系列データは画像と比較し、データを観察してアノテーションを行うことが困難である場合が多い。例えば、センサーデータを用いた行動認識では、(時刻, ラベル)の2つ組で構成される「いつ, 何をしていたか」という情報を観察者あるいは撮影した動画を利用して記録をしなければならず、一段と手間がかかる。

加えて、時系列データには本来データセットに含まれていないパターンやノイズ/異常が見られることがある。分類を行う際には、これらを与えられたラベルとして予測することは好ましくなく、その他(Void ラベル)とみなすことが望ましい。

本論文では、複数のパターンを含む時系列センサーデータのための弱教師を用いたセグメンテーション手法である LABELGEAR について述べる。

本論文で扱う問題は以下のとおりである。

問題: ウェアラブルデバイスにより収集された時系列センサーデータ集合が与えられたとき、データ中の各時刻で行われていたラベルを分類する。学習に用いるデータセットには以下の条件を満たすラベル(データラベル)が弱教師として付与されている。

- 各サンプルに付与されたデータラベルは必ず正例ラベルである。
- サンプルは1つ以上のデータラベルを取り得る。
- 本来付与されるべきだが、実際に付与されていないデータラベルが存在しても良い。

この問題に対して提案手法は、まず、(a) データラベルの分類器からデータセットに含まれる各ラベルの特徴を抽出する。ここで抽出された特徴は限られた小さなものであるが、各パターンを表現するのに重要な情報を含んでいると考えられる。これを Regime Seeds と呼ぶ。次に、(b) この Regime Seeds をデータ間で共有することにより、パターンを網羅するようさらに特徴を抽出する。そして、(c) これらの特徴を基にセグメンテーションネットワークを学習することで、各時刻に対し正確なラベルを予測する。さらに、この (b) と (c) の処理を繰り返し行うことにより、セグメンテーションネットワークを最適化する。つまり、LABELGEAR は各シーケンスに「何をしていたか」を表すデータラベルのみが与えられたデータセットに対

しセグメンテーションマスクを生成する。これはデータ収集における詳細なラベル付けを不要とし、アノテーションの手間を大きく削減する。

1.1 具体例

図1はウェアラブルセンサーを用いて収集された *Factory-worker* 2 データセットにおける LABELGEAR の処理の過程と出力結果の例を示す。このデータセットには作業員が行なった6つの作業が含まれており、図1に示すサンプルではその中から5種類の作業を計8回行っている。いつどの作業を行っていたかは図1の最下段に色と長方形の長さで表され、図上部は加速度3軸、角速度3軸で構成されるセンサー値である。LABELGEAR は4段目の最終出力を得るまでに2段目、3段目と徐々に各作業の特徴を抽出し、セグメンテーションの領域を拡大している。

図2は弱教師として“Clean(掃除)”のラベルが与えられた2つのサンプルにおける Regime Seeds と最終の出力結果を示す。ここで、データセット中の“Clean”の作業は“Wipe”と“Remove”という2つの動作から構成されている。この“Clean”の作業をどちらのサンプルもシーケンス中に含むが、図2(a), (b)の2段目が示すように、それぞれのサンプルで異なる特徴を Regime Seeds として抽出している。また、データセットでは“Rest(休憩)”というパターンが存在するが、弱教師としては付与されていない。しかし、LABELGEAR は“Rest”を“Void(図2中のピンク色)”として認識し、他のラベルと混同していない。

すなわち、LABELGEAR は、はじめに弱教師として付与されているデータラベルを分類するネットワーク (InitialNet) を学習し、Class Activation Maps(CAM) [20] を利用してデータ中のどこに注目しているのかを検出する。そして、シーケンスから切り出された小ウィンドウに対し、CAM を用いてラベルを割り当てることで Regime Seeds を生成する。Regime Seeds の領域は小さく限られたものであるが、データセットに含まれる各ラベルの重要な特徴を捉えることが可能である。次に、これらの特徴をデータ間で共有および領域拡大するため、Regime Seeds を教師とみなした小ウィンドウを分類するネットワーク (RegimeNet) を学習する。学習済みの RegimeNet は Seeds となっていない小ウィンドウのラベルを予測し、最終目標であるセグメンテーションネットワーク (SequenceNet) を学習する際の教師となる。こうして生成されるセグメンテーションマスクは初期の Regime Seeds よりも遥かに広範な領域を含んでおり、各時刻のラベルを正確に予測することができる。さらに、セグメンテーションマスクをより拡大するために、RegimeNet と SequenceNet の学習と予測を交互に繰り返し最適化を図る。ただし、これらの反復は訓練時のみであり、セグメンテーションは最終の SequenceNet で行うので効率的である。

1.2 本論文の貢献

LABELGEAR は以下の特長を持つ。

- 複数のパターンを有する時系列データに対して深層学習モデルを構築し、弱教師であるデータラベルの特徴を効果的に抽出する。

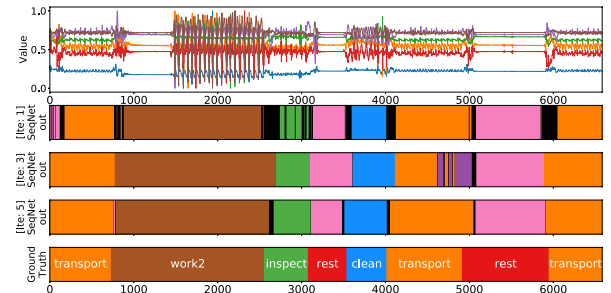
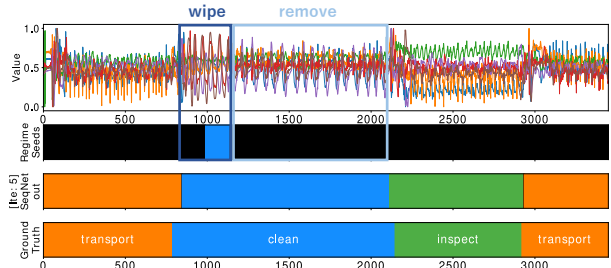
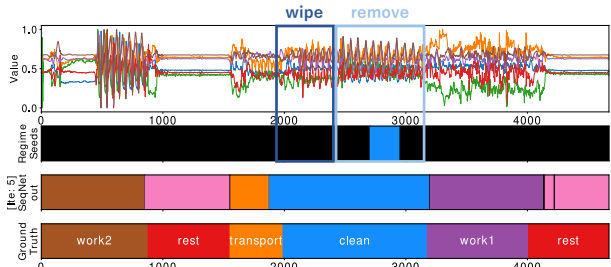


図1: イテレーションごとの LABELGEAR の出力結果の例



(a) “Clean” 中の “Wipe” が Regime Seeds となったサンプル



(b) “Clean” 中の “Remove” が Regime Seeds となったサンプル

図2: Regime Seeds と LABELGEAR の出力結果の例

- 抽出された特徴をデータセット全体で共有し、広範なセグメンテーションマスクを生成する。
- 分類の対象となるのは与えられたラベルのみであり、未知のパターンが含まれている場合はその他 (Void ラベル) として認識する。
- 学習は弱教師を用いて行われ、アノテーションのコストを削減する。

本論文の構成は以下の通りである。2章では関連研究について述べる。3章では本論文で扱う弱教師付きのデータセット、4章では提案手法、そして、5章では評価実験について述べ、6章にて本論文をまとめる。

2 関連研究

本章では、関連研究として時系列センサーデータからのパターン検出および教師ラベルを用いた分類について述べる。

2.1 パターン検出

時系列データは膨大であり、データの構造や特徴を素早く掴むことができるパターン検出やクラスタリングは重要な課題である [1], [8], [9]. Hallac ら [6] は時系列データを対象にセグメントとクラスタリングを同時に行う手法である TICC を提案し

た。TICC はマルコフ確率場に基づき、各クラスタの相互依存性を表現することにより直感的なパターンを検出する能力を持つ。しかし、クラスタ数やエラーの閾値といったパラメータ設定を必要とし、膨大な量が想定される時系列データに対しては、パラメータ設定やユーザの介入を必要としない方法が好まれる。一方、Matsubara ら [11] は隠れマルコフモデルに基づく多次元時系列シーケンスのための特徴自動抽出手法として AutoPlait を提案した。AutoPlait は最小記述長 (Minimum Description Length: MDL) の概念を用いることでデータから適切なクラスタ数を決定するため、事前知識を必要としない。

これらの手法は教師なし学習であり、データからの情報抽出においては有効であるが、特定のタスクを解くには精度の面で不十分である。本論文では、弱教師あり学習のための前処理として AutoPlait [11] を用いる。

2.2 分類問題

従来は人手による特徴量の設計が主要であったが、高次元のセンサーデータを処理することは困難であり、アルゴリズムへの悪影響やタスク依存性が乏しいことが課題として挙げられた [10], [16]。近年では、データの多次表現を獲得することができる深層学習の台頭もあり、画像認識や音声認識、自然言語処理といった分野で数多くの研究事例が報告されている [2], [14], [15]。センサーデータに対しても特徴の自動抽出は効果的であり、行動認識や心電図の不整脈分類などのモデルに組み込まれている [4], [12], [18], [19]。

しかしながら、深層学習には大量の学習用データが求められ、データの準備に多くの時間とコストを消費する。このアノテーションの効率化を図るために、画像分野では画像レベルのラベルのみからピクセル単位のセグメンテーション (Weakly-Supervised Semantic Segmentation) を行う手法が提案されている [7], [17]。また、著者ら [21] は時系列データセットから抽出したパターンに対し、Attention 機構 [3] を利用して効率的に弱教師から各パターンにラベルを割り当てた。しかし、[21] ではデータセットに弱教師が付与されているにも関わらず、教師なしの方法でパターンを抽出していた。その結果、データの性質に大きく左右され、正確なセグメンテーションを保証することができなかった。

本研究の特徴は、弱教師を用いて時系列データからの特徴抽出を行い、データ収集におけるアノテーションの負担を大きく削減しながらも、時系列データに対して高精度なセグメンテーションを実現することである。

3 弱教師付き時系列データセット

ここでは、本論文で扱う弱教師付きの時系列データセットについて述べる。

1 章でも述べたとおり、本論文では時系列データセットの各サンプルに弱教師が付与されている。弱教師は各サンプル中に含まれる時系列パターンを表すラベルであるが、その条件によって以下の種類に分けられる。

- 「最低限の弱教師」：各サンプルに 1 つずつラベルが付与

されている

- 「無欠の弱教師」：各サンプルが有する全てのパターンのラベルが付与されている

また、本研究では本来データセットに含まれていない時系列パターンにも対応すべく、データセット中にラベルとして与えられていない未知のパターンが存在するものも扱う。これを「不完全な弱教師」と呼ぶ。

4 提案手法

本章では、提案手法 LABELGEAR について述べる。

4.1 概要

データラベルの分類器は、ラベルを識別する際に注目した部分に基づき、ラベルを表現する特徴を抽出することができる。しかし、この分類器が生成する特徴は領域が限られた小さいものであるため、各ラベルのパターンを網羅的に表現するには十分でない。そこで本論文では、弱教師から抽出された領域がわずかではあるが各ラベルの識別に対して重要な特徴であるとし、この領域を徐々に拡大しつつデータセットに含まれるラベルを認識する。

本論文で提案する LABELGEAR は図 3 に示すように、ボトムアップステップとトップダウンステップの反復から構成される。まず、(1) データラベルの分類器から各ラベルの特徴 (Regime Seeds) を抽出する。次にボトムアップステップでは、(2) 得られた Regime Seeds を用いてデータ間に共通する特徴を抽出し、領域を拡大する。そして、トップダウンステップにおいて、(3) 拡大された領域からセグメンテーションネットワークを学習する。ここで得られたセグメンテーションマスクは、はじめの Regime Seeds よりも広範であり、新たな Seeds として (2) と (3) のステップを繰り返す。

こうして学習されたセグメンテーションネットワークは、データが与えられると各時刻に対して正確なラベルを予測する。つまり、(2) と (3) の反復は訓練時のみであり、予測の際には最終のセグメンテーションネットワークが適用されるので効率的である。具体的な LABELGEAR の流れはアルゴリズム 1 に示し、以降の節では、各ステップについて説明する。

4.2 Regime Seeds

ここでは、時系列データと弱教師であるデータラベルから初期領域となる Regime Seeds を抽出する方法について述べる。各ラベルの特徴を捉えるためには、データラベルを分類するネットワーク (InitialNet) を学習し、CAM [20] を使用して各ラベルのヒートマップを生成する。図 4 に CAM の出力するヒートマップの例を示す。得られるヒートマップは分類する各ラベルの部分を中心に値が高くなっており、他の部分は低くなっている。そのため、中心領域が各ラベルを識別する重要な特徴を含んでいると考えることができるが、領域は個々が小さく、粗い。そこで、時系列データをセグメントに分割し、切り出したウィンドウ毎にヒートマップの平均値を算出する。本論文では、与えられたデータに関する事前知識を必要とせず、自

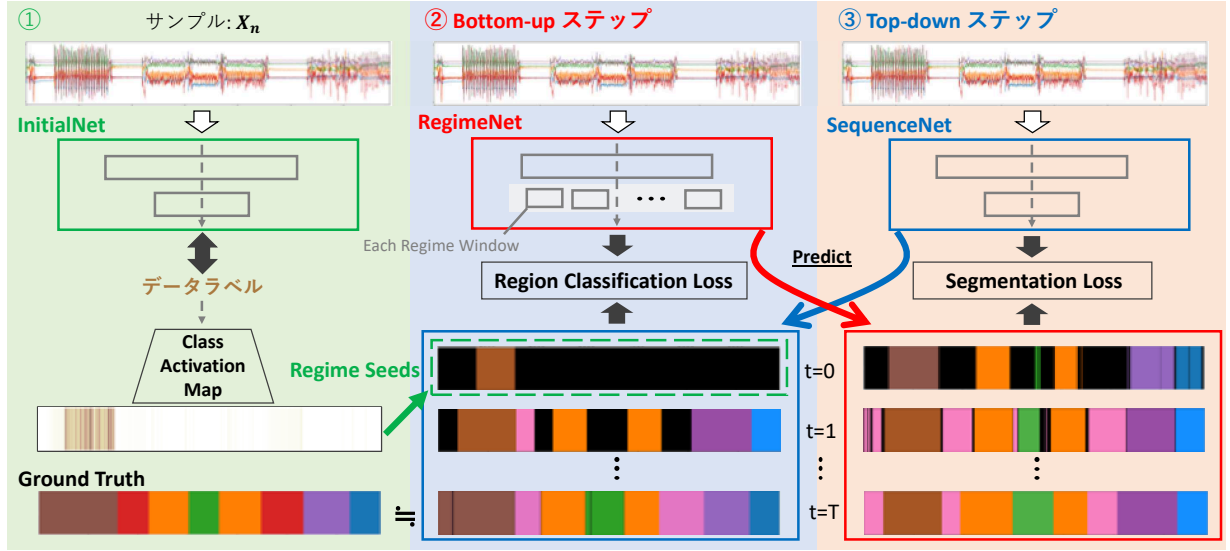


図 3: LABELGEAR のフレームワーク

Algorithm 1 Framework of the LABELGEAR

Input: Training data \mathcal{X} and small windows \mathcal{R}

Initialize: Generate initial seeds \mathcal{S} , i.e., Regime Seeds

Output: Segmentation masks \mathcal{M} and trained SequenceNet

- 1: $t = 0$
- 2: **while** iteration is effective **do**
- 3: Train the RegimeNet with \mathcal{R} and \mathcal{S}
- 4: Predict with the trained RegimeNet to get Regime \mathcal{O}
- 5: Train the SequenceNet with \mathcal{X} and \mathcal{O}
- 6: Predict with the trained SequenceNet to get segmentation masks \mathcal{M}
- 7: Update $\mathcal{S} \leftarrow \mathcal{M}$, $t \leftarrow t + 1$
- 8: **end while**
- 9: **return** \mathcal{M}

動的に時系列パターンを発見するために、AutoPlait [11] を用いる。AutoPlait は MDL に基づいてデータから適切なクラス数で分割を行うが、問題として十分に分割されない場合がある (図 5)。また、次節で述べる RoI Pooling の際に各サンプルに対し同じ数のウィンドウ数を必要とするため、本論文では大きなウィンドウから順に 2 等分し、分割を補充しつつ全データのウィンドウ数を H 個に揃える。こうして得られたウィンドウを Regime Window と呼ぶ。そして、Regime Window ごとにヒートマップの値を平均し、最大値をとるもの、および、値が閾値以上のものを Regime Seeds として抽出する。

4.3 RegimeNet

前節で抽出した Regime Seeds はセグメンテーションを行うには不十分であるが、図 4 から確認できるように各データラベルの特徴を抽出している。ここでは、Regime Seeds から領域を拡大するために、小ウィンドウごとのラベルを分類するネットワークである RegimeNet について述べる。RegimeNet は Regime Seeds からデータ間に共通する各ラベルの特徴を学

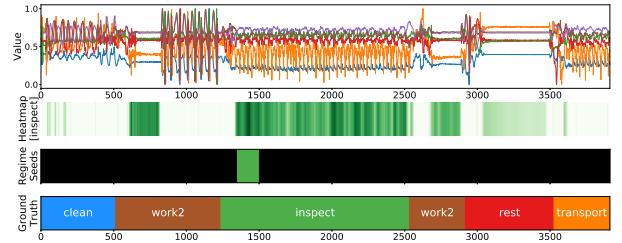
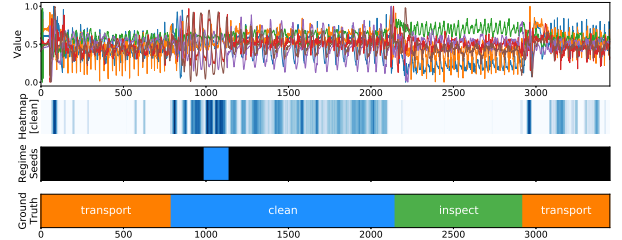


図 4: CAM の出力と Regime Seeds の例。Heatmap は色が濃いほど、高い値を示す。

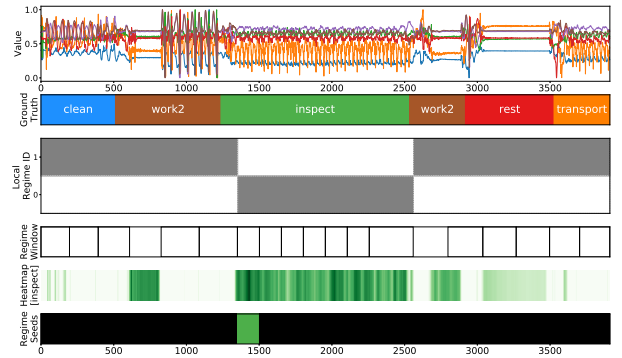


図 5: AutoPlait の分割が適切でない例: データ中には 6 つのパターンを含むが AutoPlait は 3 つにしか分割できていない (3 段目)。

習し、まだラベルが割り当てられていない小ウィンドウのラベルを予測する。

具体的には、 N 個の時系列データ集合 $\mathcal{X} = \{X_i\}_{i=1}^N$ が与えられると、まず AutoPlait を用いて分割し、各データから小ウィンドウ $\mathcal{R} = \{R_{i,j}\}_{i=1,j=1}^{N,m_i}$ を取り出す。ここで m_i は各データの Regime Seeds の数である。そして、4.2 節で得られた Regime Seeds から小ウィンドウ \mathcal{R} にラベルを付与し、 $\mathcal{S} = \{S_{i,j}\}_{i=1,j=1}^{N,m_i}$ とする。 $S_{i,j}$ は多クラス分類のため one-hot ベクトルとする。つまり、 $\mathcal{D} = \{(R_{i,j}, S_{i,j})\}_{i=1,j=1}^{N,m_i}$ を学習データとして、各データから取り出した小ウィンドウを分類する RegimeNet を学習する。このとき、RegimeNet のパラメータを θ_r とすると、小ウィンドウ $R_{i,j}$ がクラス c に属する確率は $f_c^r(R_{i,j}|\theta_r) = p(y=c|R_{i,j})$ と表現される。

RegimeNet は Fast R-CNN [5] に基づき、RoI Pooling 層を持つ (図 6)。RoI Pooling は任意のサイズの領域をプーリングして固定サイズの出力に変換する目的がある。本論文では、4.2 節で述べた Regime Window を候補領域としてデータと同時に入力し、各シーケンスの小ウィンドウをそれぞれ分類する。そして、RegimeNet の学習は以下のクロスエントロピー損失関数を最小化するように行われる。

$$\mathcal{L}_r = - \sum_{i,j,c} S_{i,j}(c) \log(f_c^r(R_{i,j}|\theta_r)) \quad (1)$$

こうして学習された RegimeNet は、データセットに含まれるラベルの共通の特徴を抽出している。すなわち、学習済みの RegimeNet を用いてラベルが不明な小ウィンドウのラベルを予測することで、領域を拡大することができる。この得られた広範な領域を Regime Region と呼び、 $\mathcal{O} = \{O_i\}_{i=1}^N$ とする。ここで重要な点として、Regime Region は Regime Seeds を基に生成されていることから、データセットにラベルとして付与されているパターンの特徴のみを捉え、そうでないパターンをその他 (Void ラベル) と認識する。したがって、LABELGEAR は時系列データ中に含まれるノイズや異常パターンを分類ラベルとは別のラベルとして識別することができる。

4.4 SequenceNet

ここでは、前節で生成したより広範な Regime Region を教師とみなし、本研究の最終目標であるセグメンテーションネットワークである SequenceNet を学習する。本論文では、「最低限の弱教師」のように各時系列データが有する全てのパターンの弱教師が付されていない場合においても、RegimeNet による補完によりデータに含まれる共通な特徴を抽出できる。しかしながら、より正確なセグメンテーションを達成するには「無欠の弱教師」が付与されているデータセットが好ましく、Regime Region を生成する際に誤ったラベルを取り除くことができる。この場合では、本節での学習において教師に正しいラベルのみが含まれるようになり、生成されるセグメンテーションマスクはより堅牢なものになる。

SequenceNet は N 個の時系列データ集合 $\mathcal{X} = \{X_i\}_{i=1}^N$ と 4.3 節で得られた Regime Region $\mathcal{O} = \{O_i\}_{i=1}^N$ が教師として与えられると、セグメンテーションマスクを生成するよう学習する。このとき、SequenceNet のパラメータを θ_s とすると各時系列データの時刻 t がクラス c に属する確率は

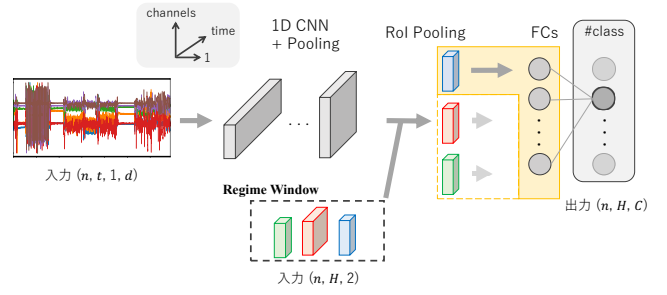


図 6: RegimeNet の構造

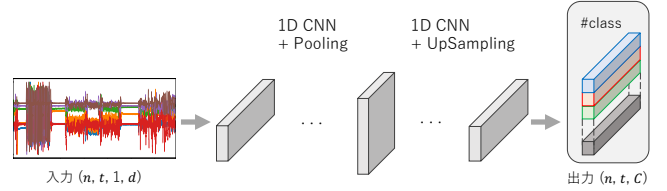


図 7: SequenceNet の構造

$f_{t,c}^s(X|\theta_s) = p(y_t = c|X)$ と表現され、以下のクロスエントロピー損失関数を用いる。

$$\mathcal{L}_s = - \frac{1}{\sum_{c=1}^C |S_c|} \sum_{c=1}^C \sum_{t \in S_c} \log(f_{t,c}^s(X|\theta_s)) \quad (2)$$

C は分類するクラスラベルの数であり、 $|S_c|$ は教師としてクラス c が割り当てられた時刻の数である。

RoI Pooling を用いて小ウィンドウを分類していた RegimeNet では、特徴として各ウィンドウ内のみを考慮していた。一方、SequenceNet では図 7 に示すように時系列データ全体を含み、前後の関係を特徴として捉える。

本論文では、SequenceNet の出力を新たな Seeds として RegimeNet に与え、4.3 節と本節の処理を繰り返す。この反復では、学習に用いる領域を拡大し、データセットから各ラベルを表現する特徴をより堅牢にする目的がある。そして、抽出された特徴はセグメンテーションネットワークを徐々に最適化し、不明であったパターンを認識しながら、正確なセグメンテーションを行う。次章では、最終的に学習された SequenceNet を使用して、LABELGEAR の評価実験を行う。

5 評価実験

本章では、LABELGEAR の有効性を検証するため、実データを用いた実験について述べる。まず、5.1 節では提案手法の精度の評価および既存手法との比較をし、さらに 5.2 節では、反復アルゴリズムの有効性を検証するために以下の 3 つの観点からアブレーションテストを行った。

- Q1 イテレーションごとの各ネットワークの精度の推移
- Q2 反復を 1 つのネットワークで行なった場合との精度の比較
- Q3 弱教師の条件を変えた場合との精度の比較

実験には以下のデータセットを用いた。

- HHAR: 本データセットは、Stisen ら [13] が 9 人のユーザーに対する 6 種類のアクティビティ (bike, sitting, standing,

walking, climbStair-up, climbStair-down) の加速度 (3 軸), 角速度 (3 軸) をスマートフォンを用いて収集したものである。本実験では, ユーザ b の bike, standing, walking の 3 種類を用い, ランダムに 2 つ以上を組み合わせるシーケンスを作成した。

- *Factory-worker*: 本データセットは, 工場における作業員を模した作業である Polish(表面を磨く), Wipe(布で拭く), Clean(ゴミを取り除く) の 3 種類の動作を腕に装着して加速度 (3 軸), 角速度 (3 軸) を計測することができるアームバンド型の Myo¹ を用いて収集されたものである。HHAR と同様にシーケンスの作成を行なった。
- *Factory-worker 2*: 本データセットは, 工場における作業員を模した作業である Clean(掃除), Transport(運搬), Inspect(点検), Rest(休憩), Work1(組み立て), Work2(工作) の 6 種類の動作を *Factory-worker* と同様に Myo を用いて収集されたものである。

5.1 提案手法のセグメンテーション精度

はじめに, 提案手法のセグメンテーションの精度を検証するため, 既存手法 [21] と以下の比較手法を用いて比較を行なった。

- ベースライン: 各時系列に付与されているデータラベルを全ての時刻に付与する
- Kmeans: 時系列から Sliding Window ごとの各種統計量 (平均, 中央値, 分散, 標準偏差) を特徴とし, Kmeans でクラスタリングを行うことによって各時刻にラベルを付与する

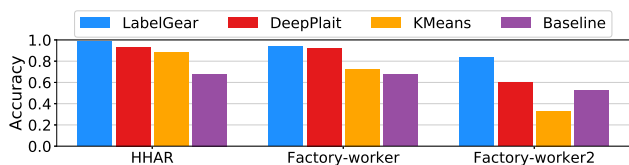
精度の評価には正解率 (Accuracy) と mIoU(mean IoU) の 2 つの指標を用いた。

- 正解率 (Accuracy): 時系列中の正解ラベルと等しいラベルが割り当てられている時刻をカウントし, 時系列の長さで割る
- mIoU(mean IoU): IoU (Intersection over Union) を算出し, ラベルごとの IoU の平均値をとる

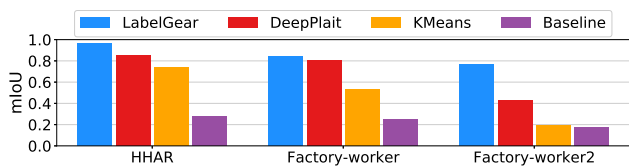
また, 各データセットに付与した弱教師の条件は以下のとおりである。

- HHAR, *Factory-worker*: 「最低限の弱教師」
- *Factory-worker 2*: データ中に含まれる “Rest(休憩)” のラベルを与えない 「不完全な弱教師」 かつ 「最低限の弱教師」

図 8 は, 提案手法と既存手法 [21], 上記の比較手法との精度の比較である。提案手法は全てのデータセットにおいて優れた性能を示している。特に, *Factory-worker 2* において [21] は正解率 59.98%, mIoU 42.49% の精度にとどまったが, LABELGEAR は正解率 83.20%, mIoU 77.06% と大きく上回った。これにはまず, [21] が弱教師を利用せずに時系列データを分割しており, 各パターンの特徴を十分に識別することができなかったことが挙げられる。一方, LABELGEAR は弱教師を用いて分割対象の時系列パターンを表現する特徴を効果的に抽出するため, 高精度なセグメンテーションを達成した。この結果から, LABELGEAR はあらゆるデータセットに対し堅牢なセグメンテーションマスクを生成することができるといえる。



(a) 正解率 (Accuracy)



(b) mIoU (mean IoU)

図 8: 各データセットにおけるセグメンテーション精度の比較

表 1: *Factory-worker 2* における各ネットワークの正解率の推移

		train	test
Regime Seeds		27.86%	-
ite1	RegimeNet	75.20%	-
	SequenceNet	70.44%	65.25%
ite2	RegimeNet	84.53%	-
	SequenceNet	80.45%	76.70%
ite3	RegimeNet	85.52%	-
	SequenceNet	84.98%	80.43%
ite4	RegimeNet	86.51%	-
	SequenceNet	87.01%	82.49%
ite5	RegimeNet	87.03%	-
	SequenceNet	87.38%	83.20%

5.2 アブレーションテスト

続いて, *Factory-worker 2* を用いて LABELGEAR の反復アルゴリズムの有効性を検証した。

Q1. イテレーションごとの各ネットワークの精度の推移

ここでは, 提案手法の特徴抽出とセグメンテーションネットワークの学習の有効性を検証するため, イテレーションごとの RegimeNet と SequenceNet の精度を評価する。表 1 に *Factory-worker 2* における正解率の推移を示す。InitialNet に CAM を適用して生成した Regime Seeds は正解率が 27.86% と値が小さく, 時系列パターンを捉える特徴としては粗い。しかし, RegimeNet でそれぞれのシーケンスから抽出した特徴を共有することで正解率は 75.20% にまで上昇し, イテレーション 1 におけるセグメンテーションマスクは 70.44% となっている。そして, イテレーションが経過するにつれてセグメンテーションマスクは徐々に拡大し, 最終的には *Factory-worker 2* のテストデータに対して正解率が 83.20% となった。したがってこの結果から, 弱教師を利用して抽出した特徴からセグメンテーションネットワークを学習する反復アルゴリズムの有効性が確認できる。

Q2. 反復を 1 つのネットワークで行なった場合との精度の比較

ここでは, さらに提案手法のアルゴリズムの有効性を検証するため, RegimeNet と SequenceNet の交互の繰り返し学習を行わない ONEGEAR との精度を比較する。ONEGEAR は 1 回目の SequenceNet の学習までは LABELGEAR と同様であるが, それ以降は RegimeNet を用いずに, SequenceNet のみで

¹: <https://support.getmyo.com/hc/en-us>

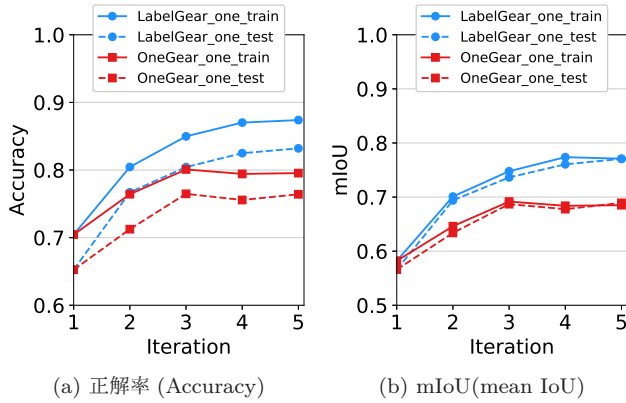


図 9: *Factory-worker 2* における LABELGEAR と ONEGEAR の精度の推移

反復を行う。図 9 に *Factory-worker 2* における LABELGEAR と ONEGEAR の精度の推移を示す。ONEGEAR はイテレーションの経過による精度の上昇が緩やかであり、最終的なセグメンテーション精度はテストデータにおいて正解率が 76.41% となった。一方で、LABELGEAR は ONEGEAR と比較して精度の上昇が早く、イテレーション 2 の時点で ONEGEAR の最終の精度を上回っている。そして、その後も精度は向上し、最終的にテストデータにおいて正解率が 83.20% に達した。この結果が得られた要因は、1 つ前のセグメンテーションマスクから生成した Seeds を RegimeNet に与えることより、信頼性の高い特徴を用いて領域の拡大を行えたからだと考えられる。したがって、LABELGEAR はイテレーションごとにセグメンテーションネットワークを最適化し、堅牢なセグメンテーションマスクを生成できるといえる。

Q3. 弱教師の条件を変えた場合との精度の比較

ここでは、データセットに付与する弱教師の条件に対する LABELGEAR の性能を評価する。これまでの実験で使用した *Factory-worker 2* の弱教師は以下の要件を満たしていた。

1. **one** : データ中に含まれる“Rest(休憩)”のラベルを与えない「不完全な弱教師」かつ「最低限の弱教師」

本節では、以下の 2 つの弱教師の条件と比較を行う。

2. **all** : データ中に含まれる“Rest(休憩)”のラベルを与えない「不完全な弱教師」かつ「無欠の弱教師」

3. **perfect** : “Rest(休憩)”のラベルも与えた「無欠の弱教師」

弱教師の条件を変えた場合の *Factory-worker 2* における LABELGEAR の正解率の比較を図 11 に示す。前節の条件 1 (“one”) では、テストデータに対し正解率 83.20% であった (図 11 左) が、条件 2 (“all”) の場合は正解率が 89.04% と上昇した。また、図 10 は条件 2 における LABELGEAR と ONEGEAR のイテレーションごとの精度の推移を示すが、1 回目のイテレーション終了時に正解率は 80% を超え、ONEGEAR でさえも最終的に条件 1 の LABELGEAR よりも優れている。このことから、LABELGEAR の性質として学習に用いるデータの数および付与されているラベルの数が多いほど多様な特徴を発掘す

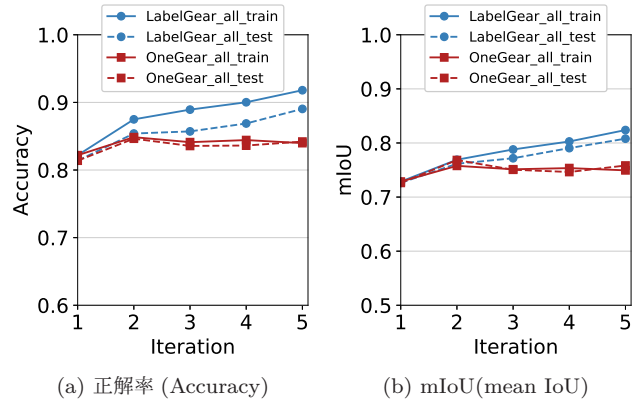


図 10: *Factory-worker 2* の条件 2 における LABELGEAR と ONEGEAR の精度の推移

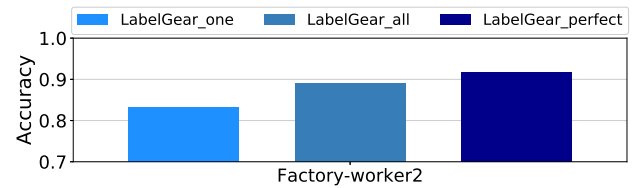


図 11: *Factory-worker 2* における弱教師の条件による LABELGEAR の正解率の比較

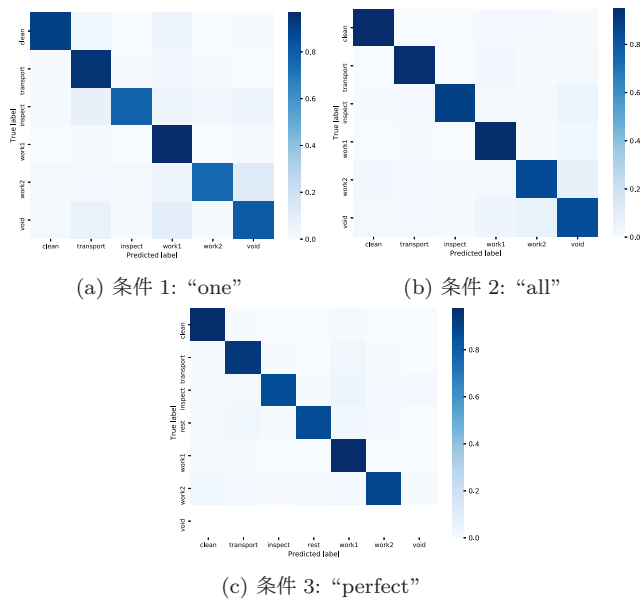


図 12: *Factory-worker 2* における各条件の混同行列

ることができ、モデルの性能が上がるといえる。また、「無欠の弱教師」には「誤ったラベルが付与されていない」かつ「全てのラベルが付与されている」という条件があり、学習の過程で不正確なラベルの特徴を取捨選択が可能なのも精度向上の要因であると考えられる。しかしながら、条件 1 の LABELGEAR が条件 2 の ONEGEAR と同等である結果からも、2 つのネットワークの組み合わせの有効性が確認できる。そして、条件 3 (“perfect”) では “Rest” も含めた全てのラベルを与えた。正解率は 91.69% とさらに上昇したが、条件 2 においても十分に “Rest” を “Void” として認識し、他のクラスと識別することが

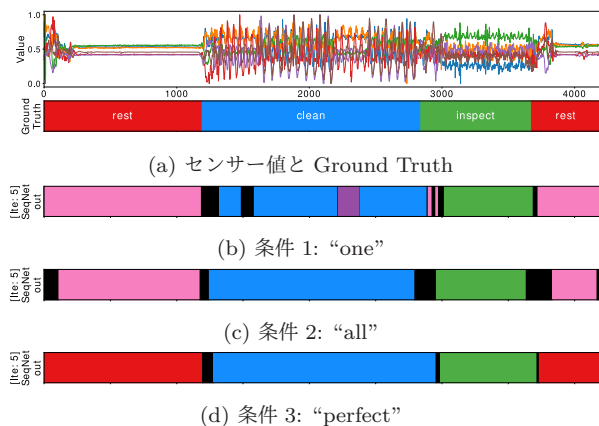


図 13: *Factory-worker 2* における各条件の出力結果の例

できたといえる。図 12 は各条件における混同行列を示す。条件 3 においても“Void”への誤予測が確認できるのは、Ground Truth において作業と作業の間の時間を考慮しておらず、その境界を“Void”と予測している結果である。

最後に、*Factory-worker 2* における各条件の出力結果の例を図 13 に示す。図 13(b), (c) では、“Rest(赤色)”を“Void(ピンク色)”とみなして予測をしているが、図 13(d) では“Rest”のラベルも学習に用いているため“Rest”は赤色となっている。

6 む す び

本論文では、複数のパターンを含む時系列センサーデータのための弱教師を用いたセグメンテーション手法である LABELGEAR を提案した。LABELGEAR は、各時系列データに「何をしていたか」というラベルのみが記録されたデータ集合からデータセットに含まれるパターンの特徴を抽出し、2つのネットワークを用いた反復的なアルゴリズムによって徐々に認識する特徴を拡大する。その際、データセットにラベルとして付与されていないパターンはその他ラベルとして区別される。実データを用いた評価実験では、提案手法が既存手法と比較して高精度を達成し、弱教師から効率的に学習を行うことができることを確認した。これはラベル付きの学習用データの収集にかかるコストを大きく削減することに貢献する。

文 献

- [1] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering—a decade review. *Information Systems*, 53:16–38, 2015.
- [2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International Conference on Machine Learning*, pages 173–182, 2016.
- [3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ICLR*, 2015.
- [4] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
- [5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE inter-*

- national conference on computer vision*, pages 1440–1448, 2015.
- [6] D. Hallac, S. Vare, S. Boyd, and J. Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 215–223. ACM, 2017.
- [7] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.
- [8] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and information systems*, 8(2):154–177, 2005.
- [9] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM’05)*, pages 8–pp. Ieee, 2005.
- [10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.
- [11] Y. Matsubara, Y. Sakurai, and C. Faloutsos. Autoplait: Automatic mining of co-evolving time sequences. In *SIGMOD*, pages 193–204, 2014.
- [12] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.
- [13] A. Stisen, H. Blunck, S. Bhattacharya, T. S. Prentow, M. B. Kjærgaard, A. Dey, T. Sonne, and M. M. Jensen. Smart devices are different: Assessing and mitigating mobile sensing heterogeneities for activity recognition. In *ACM Conference on Embedded Networked Sensor Systems*, pages 127–140. ACM, 2015.
- [14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [16] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11, 2019.
- [17] X. Wang, S. You, X. Li, and H. Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1354–1362, 2018.
- [18] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy. Deep convolutional neural networks on multi-channel time series for human activity recognition. In *IJCAI*, pages 3995–4001, 2015.
- [19] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *WWW*, pages 351–360, 2017.
- [20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [21] 入船恭彰, 松原靖子, 川畑光希, 山室冨, 浦晃, 河東孝, 梅田裕平, 小橋博道, 稲越宏弥, 櫻井保志. 弱ラベルを用いた時系列データへの詳細ラベル付与. In *DEIM Forum*, 2019.