

## 推薦論文

# 移動手段推定のための表現学習を用いた GPS ログからの特徴抽出

遠藤 結城<sup>1,a)</sup> 戸田 浩之<sup>1,b)</sup> 小池 義昌<sup>1,c)</sup>

受付日 2015年3月21日, 採録日 2015年7月1日

**概要:** 本稿では GPS 測位などによって取得されたユーザの移動履歴 (GPS ログ) から, 歩行や車などのユーザの移動手段を推定する移動手段推定に取り組む. 従来の移動手段推定手法は, 複数ユーザの GPS ログと移動手段アノテーションを用いることで, 教師あり学習の枠組みで推定モデルを構築している. 教師あり学習を用いるアプローチにおいては, GPS ログから推定に有効な特徴を抽出することが大きな課題であり, 従来は試行錯誤を通じて人手によって特徴が設計されてきた. しかしながら, 人々の行動は多様であり, 移動軌跡も様々な特徴を含んでいるため, 人手による特徴設計ではすべての特徴を網羅するような特徴抽出が困難であるという限界がある. 我々は, 特徴表現をデータから自動的に獲得する表現学習を用いることで, 人手による特徴設計では困難な推定精度達成が可能であると考えた. 本稿では, 高性能な表現学習を実現するために, 入力された GPS ログから移動手段に特徴的な情報を保持したまま表現学習の入力に変換する手法を開発し, これを採用したフレームワークを提案する. 提案フレームワークにおいては, 表現学習として deep learning を用いて新たな特徴表現を抽出し, これを加えた特徴をもとに教師あり学習を行うことで推定モデルを構築する. 実データである GeoLife データセットを用いた評価実験の結果, 人手による特徴設計を用いる従来手法と比べて, 提案手法が高い推定精度を示した. これにより, GPS ログに対する表現学習の有効性を確認した.

キーワード: 移動軌跡, 移動手段, 表現学習, deep learning

## Feature Extraction from GPS Logs Using Representation Learning for Transportation Mode Estimation

YUKI ENDO<sup>1,a)</sup> HIROYUKI TODA<sup>1,b)</sup> YOSHIMASA KOIKE<sup>1,c)</sup>

Received: March 21, 2015, Accepted: July 1, 2015

**Abstract:** This paper tackles the problem of estimating user's transportation modes such as walking and car from their movement trajectories (GPS logs). Existing research has adopted supervised learning approaches and constructed an estimation model. In such approaches, creating effective features for estimation from GPS logs is a challenging problem and researchers have used their ingenuity to find effective features by trial and error. However, such handcrafted features cannot cover all features for ideal estimation because people's behaviors are diverse and movement trajectories also include various aspects. We focus on representation learning that can automatically learn features and this may enable more accurate estimation of transportation modes. To achieve this, we have developed a method that can convert raw GPS logs into image data while keeping discriminative information for representation learning, and we propose a framework for transportation mode estimation that includes this method. In our framework, we adopt a deep learning for extracting new feature representations. An estimation model is then constructed in a supervised manner using the extracted features. From the results of evaluation using a GeoLife dataset, our technique achieves higher accuracy than feature engineering approaches and we confirm effectiveness of representation learning for trajectory data.

**Keywords:** movement trajectories, transportation mode, representation learning, deep learning

<sup>1</sup> 日本電信電話株式会社 NTT サービスエボリューション研究所  
NTT Service Evolution Laboratories, NTT Corporation,  
Yokosuka, Kanagawa 238-0847, Japan

a) endo.yuki@lab.ntt.co.jp

b) toda.hiroyuki@lab.ntt.co.jp

c) koike.y@lab.ntt.co.jp

本稿の内容は 2014 年 11 月の WebDB フォーラム 2014 にて発表され, 同シンポジウムプログラム委員会により情報処理学会論文誌データベースへの掲載が推薦された論文である.

## 1. はじめに

Google Now<sup>\*1</sup>や Moves<sup>\*2</sup>をはじめとした、スマートフォンなどから取得されたユーザの位置情報を利用したサービスが増加している。そのような位置情報を利用したサービスの国内市場規模は、2012年度は19.8兆円であるところ2020年度には62.2兆円になるとも予測されている<sup>\*3</sup>。それにともない端末が利用可能なユーザの位置情報を利用して、ユーザ状況を推定する技術の必要性が高まっている。位置情報に基づくユーザ状態推定の研究としては、ユーザが訪問した地域を推定する滞在推定 [15] や、何の移動手段を用いたかという移動手段推定 [25], [27] などの研究が行われてきた。特に移動手段推定の技術によって、ユーザの旅行時などのライフログや、移動手段に応じて適切なサービスを提供するパーソナルアシスタントが可能になると考えられる。このことから、本稿では位置情報を用いたユーザ状態推定の中でも、移動手段推定に着目する。

移動手段推定とは、GPS 測位などにより取得された位置情報と時刻情報を持つ測位点系列 (GPS ログ) に対して、(1) 同じ移動手段を用いて移動していた区間 (セグメント) の抽出と、(2) セグメントに対する移動手段 (たとえば歩行、自動車、電車など) の推定を行うものである。セグメント抽出 (1) については、一定時間で区間を区切る方法や GPS ログから得られる速度、加速度情報を手がかりとした変化点に基づく方法 [25], [27] などがある。セグメントに対する移動手段推定 (2) については、GPS ログに対する移動手段アノテーションを用いることで、教師あり学習の枠組みで移動手段推定モデルを構築する方法 [25], [27] が利用されている。

教師あり学習を用いた移動手段推定においては、単純な測位点の羅列からなる情報から、推定に有効な特徴をいかに用意できるかが推定精度向上において重要な課題となる。データから特徴を抽出するアプローチは大きく2つに分けられる。

1つ目のアプローチは人手による特徴設計 (**feature engineering**) である。Feature engineering とは、試行錯誤を通じてタスクに応じて推定に有効な特徴を設計する方法である。たとえば移動手段推定の従来研究 [25] は、GPS ログから移動距離や速度、加速度などの特徴を設計し、その特徴をもとに推定モデルを構築している。さらにその後発展的な特徴 [26], [27] が追加され、推定精度の向上が確認されている。

2つ目のアプローチは**表現学習**である。表現学習はデータを表す特徴から、データをよく表現する高次の特徴を新

たに獲得する手法の総称である。代表的な手法として、近年画像認識や音声認識の分野において抜本的な精度向上を達成した**深層学習 (Deep Learning)** [2] があげられる。Deep learning は従来よりも多層のニューラルネットワーク (Deep Neural Network; DNN) を学習し、その中間層を高次の特徴と見なすことで、高い表現力を持つ特徴 (中間表現) を獲得することができる。しかしながら、データからどのような中間表現を取得できるのか、得られた中間表現がタスク性能に寄与するのかということはタスク依存の側面が大きい。画像や音声と異なる GPS ログなどのデータに対して、deep learning を適用すること自体が従来ほとんど検討されていないことであり、他分野への応用が課題になっている。

我々は、移動手段推定などの GPS ログを対象にしたユーザ状態推定のタスクにおいては、feature engineering に限界があり、表現学習を用いることでその限界を突破できると考えた。feature engineering ではすべての特徴を網羅するような特徴抽出が困難だと考えられる。なぜならば人々の行動は多様であり、同じ移動手段であってもユーザによって移動距離や速度が異なるなど、移動軌跡は様々な特徴を含んでいるからである。表現学習を用いることで、GPS ログから feature engineering では得られない有効な特徴を抽出でき、これにより移動手段推定の精度を向上させることが可能であると考えられる。

本稿では deep learning を用いた表現学習により、GPS ログから特徴を抽出する手法を提案し、それを移動手段推定の枠組みに組み込む。我々のキーアイデアは、画像認識の分野において deep learning が成功を収めているという知見のもと、GPS ログを軌跡画像として表現することである。この際、効果的に表現学習が行えるように、移動手段推定に有効な情報を多く残すように軌跡画像を生成する。そのような軌跡画像の生成にはいくつかの技術的課題がある。まず、deep learning の手法は基本的に入力データが同次元である必要がある。しかし、入力となるセグメントの地理的範囲は移動手段によって異なる (たとえば歩行は狭く、自動車は広くなる) ため、単純にセグメントを包含する軌跡画像を生成しても、画像サイズが異なり入力に用いることができない。次に、経緯度という高精細な単位で表現された GPS ログを画素単位に量子化する必要がある。画像サイズを小さくすると、1画素に対応する地理的範囲が大きくなり、軌跡の情報が失われてしまう。一方で画像サイズを大きくすると、全画素に対して GPS 軌跡に関わる情報を持つ画素が少ない画像となり、効果的な表現学習が難しくなる。本研究ではこれらの課題を達成するために、GPS ログに含まれる特徴的な情報を保持したまま軌跡画像を生成する手法を提案する。

軌跡画像を生成した後は、deep learning の手法である Stacked Denoising Autoencoder (SDA) [22] によって高次

\*1 <http://www.google.com/landing/now/>

\*2 <https://play.google.com/store/apps/details?id=com.protogeo.moves>

\*3 出典：総務省「G空間×ICT推進会議」(第4回)資料

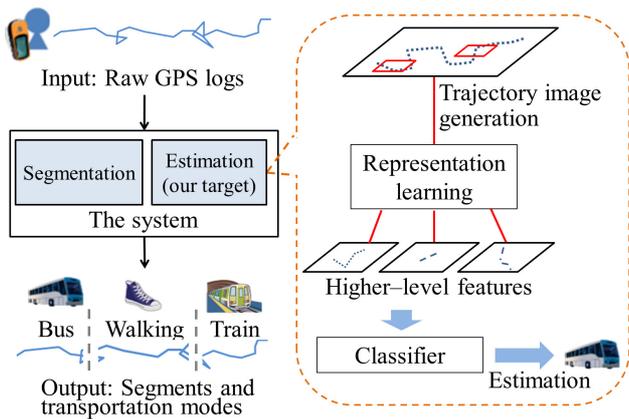


図 1 移動手段推定の入出力および提案手法の概要. 移動手段推定は GPS ログを入力として, セグメントおよびセグメントに対する移動手段を推定する. 移動手段の推定には, 軌跡画像化した GPS ログをもとに, deep learning によって学習した DNN を用いて抽出された特徴を用いる

Fig. 1 Input and output of transportation mode estimation and overview of proposed method. Transportation modes on each segment are estimated using GPS logs as input. We use deep features extracted from trajectory images of GPS logs using DNN trained by deep learning.

の特徴を抽出し, これを移動手段推定における教師あり学習に利用する. 移動手段推定の基本的な問題設定は Zheng ら [25], [27] の研究を踏襲し, 特徴抽出および教師あり学習の枠組みを拡張する. 移動手段推定の問題設定と本稿の貢献を図 1 に示す. 入力として GPS ログを与えると, 出力として GPS ログのセグメントおよび各セグメントに対する移動手段が得られる. 移動手段は deep learning によって学習した DNN に基づいて抽出された特徴から推定される.

本研究の貢献をまとめると以下ようになる.

- GPS ログからの軌跡画像生成手法の提案
- Deep learning を用いた軌跡画像からの表現学習と移動手段推定への応用のフレームワークの提案
- 実データを用いた実験による提案手法の有効性の検証

## 2. 関連研究

本章では GPS ログを用いたユーザ状態推定の関連研究を述べた後, deep learning について簡潔に説明する.

**GPS ログを用いたユーザ状態推定.** GPS ログを用いた研究として, ユーザ活動状態推定 [3], [6], [13], [16], 移動手段推定 [12], [17], [25], [27], 地域分析 [23], [24] などがあげられる. 特に移動手段推定と関連の深い研究として, GPS ログに加えセンサから取得した体温や心拍数, 湿度や光量などのデータを特徴として用い推定モデルを構築することで, 歩行やランニング, サイクリング, ボートなどのユーザの行動を推定する手法が提案されている [6], [16]. これらの手法は, 様々なウェアラブルセンサを用いて取得した

データを利用することで, 多様なユーザの行動推定が可能である. しかしながら, 一般ユーザが日常生活において様々なセンサを携帯するのは負担が大きいと, 理想的には少量のセンサから得られたデータでユーザの状態を推定できることが望ましい. そのため, GPS ログ以外の情報を用いる方法は本稿の範疇外とする.

これに対し, Liao ら [12] や Patterson ら [17] は, センサデータとして GPS ログのみを用いて歩行やバス, 車などの移動手段を推定した. しかしこれらの手法は, バス停や駐車場の情報を含むストリートマップなどの補足となる地理情報を必要とする. 実世界の街の構造は時間とともに変化することを考えると, ストリートマップのような静的な情報では対応できなくなる可能性がある. そのため, 本稿では特定の外部情報を必要としない方法を対象とする.

外部情報を使わないアプローチとして, Zheng ら [25], [27] による GPS ログのみから移動手段を推定する手法があげられる. 彼らは測位点から得られる速度と加速度を用いた移動手段の変化点検出によって, GPS ログのセグメント分割を行う手法を提案している. そして分割された各セグメントに対するマルチクラス分類問題として, 移動手段推定を実現している. この手法では分類器の構築に用いる特徴を手で設計している. 特徴には移動距離や速度, 加速度など基本的なものがはじめに提案された [25]. その後さらに, 一定区間において大きな速度変化のある測位点の割合 (Velocity Change Rate; VCR), 停止している測位点の割合 (Stop Rate; SR), 大きく進行方向を変えている測位点の割合 (Heading Change Rate; HCR) など発展的なものが提案され, 推定精度の向上が示されている [26], [27]. これに対し本研究は, deep learning の枠組みで GPS ログを表現した画像から, 人手での設計が困難な特徴を抽出する課題に取り組む点で既存研究と異なる.

**Deep learning.** Deep learning の主な目的は, 信号レベルの入力から DNN を学習することで, 高次の表現を獲得することにある. 従来の画像認識タスクにおいては, たとえば SIFT や HOG 特徴などに代表される, 認識に効果的な特徴を手で設計する feature engineering が主流であった. 一方で DNN は生の画像データそのもの, つまり画像の各画素値のベクトル表現を入力するだけで, 認識に有効な特徴を自動で抽出できると考えられている. DNN によって実際に得られた特徴と画像データに付与されたクラスラベルを用いて教師あり学習を行うことで, 高い認識精度を達成できたことが示されている [7], [8].

しかしながら, 従来 DNN は多層のネットワークから構成されることからの表現力の高さゆえに, 誤差逆伝播法 (back-propagation) によって出力層から学習を行うと, 誤差情報が入力層付近まで伝播しきらず途中で消失してしまい, 過学習しやすいという欠点があった. この問題を克服するため, 2006 年に greedy layer-wise training と呼ばれ

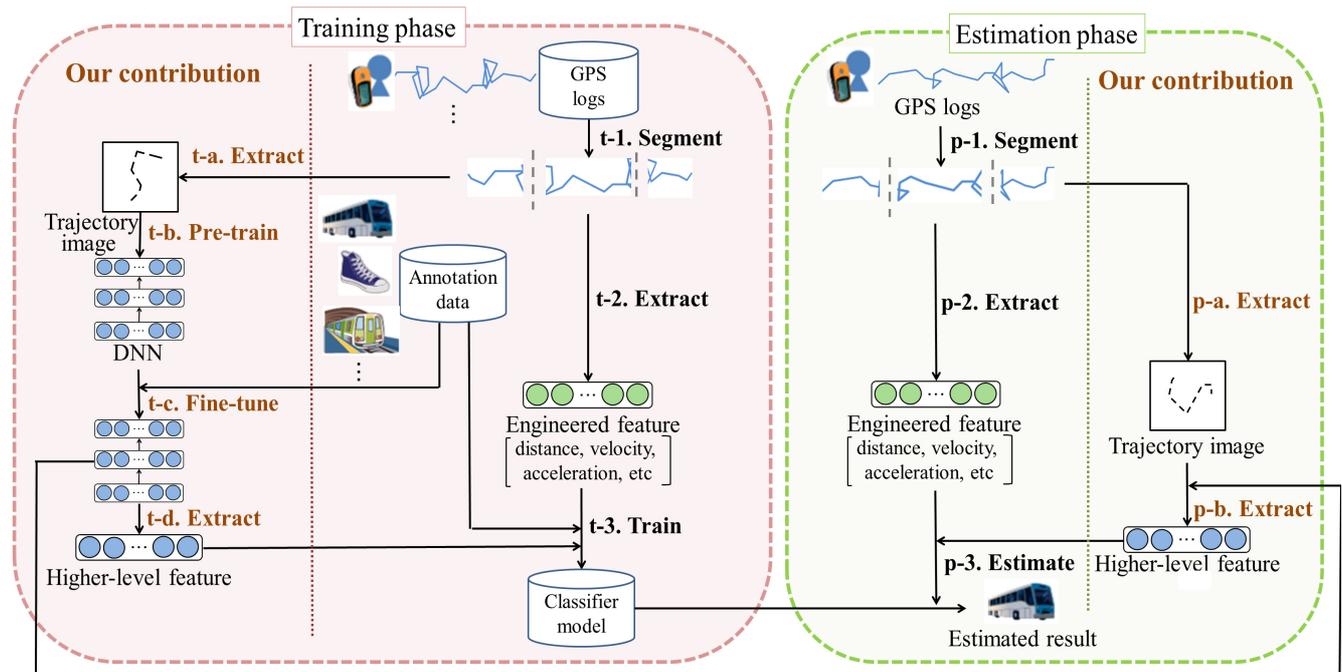


図 2 本稿で提案する移動手段推定システムの処理の流れ

Fig. 2 Processing flow of proposed system of transportation mode estimation.

る DNN の学習手法が登場し、これが deep learning が注目される 1 つの要因となった [1], [7], [8]. Greedy layer-wise training は教師ありでネットワーク全体のパラメータを学習する前に、入力層側から 1 層ずつ中間層のパラメータを教師なしで調整する事前学習 (pre-training) を行う方法である。これをネットワーク全体の初期パラメータとして利用し、その後教師あり学習で全体のパラメータを調節 (fine-tuning) することで出力層からの誤差情報も十分に行きわたらせることが可能になり、結果としてこれが様々なタスクに対する精度の向上へとつながった。以来、deep learning は画像認識 [9], [11] だけでなく、音声認識 [5] や自然言語処理 [4], [18] の分野にも多くの応用がなされるようになり、目覚ましい成果をあげている。

Deep learning の関連手法としては、Deep Belief Nets (DBN) [7] や、Deep Boltzmann Machine (DBM) [19], SDA [22] などがあげられる。その他の手法についてはサーベイ論文 [2] を参照されたい。本研究では特に SDA に注目する。SDA は前述のとおり多層のニューラルネットワークの pre-training として用いられ、画像認識へも応用されている。本研究では GPS ログを表す軌跡画像を入力とし、SDA による pre-training および、正解ラベルに基づく fine-tuning を行うことで DNN を学習し、移動手段推定など GPS ログを入力とした分類タスクにおいて有効な特徴の自動抽出を試みる。

### 3. 教師あり学習を用いた移動手段推定

教師あり学習を用いた GPS ログに対する移動手段推

定システムの処理の流れを図 2 に示す。移動手段推定の処理は訓練フェーズ (training phase) と、推定フェーズ (estimation phase) の 2 つに分けられる。

まず、推定モデルを構築する訓練フェーズについて述べる。与えられた各ユーザの GPS ログ情報について、移動手段の切り替わりに基づくセグメントを抽出する (t-1)。ここで得られたセグメントを学習や推定における 1 事例として扱う。セグメント分割手法には Zheng ら [25], [27] の手法を用いることを想定しており、既存手法が所与のものとして本稿では範疇外とする。次にセグメント分割された訓練データから特徴を抽出する (t-2)。特徴の抽出には、従来の feature engineering による方法と表現学習による方法の両方を用いる。前者については、Zheng ら [25], [27] と同様の特徴 (engineered feature) として、セグメント長、平均速度、速度の期待値、速度の分散、当該セグメントにおける上位 3 件の速度と加速度、VCR, SR, HCR の計 13 次元の特徴を抽出する (t-2)。後者については、表現学習を適用するために、まず各セグメントにおける GPS ログから軌跡画像を生成する (t-a, 3.1 節)。次に、抽出された軌跡画像を用いて DNN を学習する (3.2 節)。DNN の学習には pre-training として SDA を用いる (t-b)。Pre-training の後は、アノテーションも利用して DNN 全体のパラメータを調節する fine-tuning を行う (t-c)。最後に、学習した DNN を用いて高次の特徴を抽出する (t-d)。以上の処理によって得られた 2 種類の特徴を連結し、これらを用いて教師あり学習の枠組みでマルチクラス分類器を生成する (t-3)。

推定フェーズにおいては、訓練フェーズと同様にGPSログをセグメント分割し (p-1)、セグメントごとに engineered feature の抽出 (p-2) と軌跡画像の抽出 (p-a) を行う。軌跡画像は訓練フェーズで学習した DNN を用いて高次の特徴に変換する (p-b)。得られた2つの特徴を連結し、訓練フェーズで生成した推定モデルを用いて各セグメントに対する移手段推定結果を出力する (p-3)。

以降ではGPSログからの軌跡画像の生成手法を述べたのちに、SDAを用いた軌跡画像からの特徴抽出方法を述べる。

### 3.1 GPSログからの軌跡画像の生成

GPSログを画像として表現するには2つの問題点があげられる。1つ目は、セグメントごとにGPSログの地理的範囲が異なるということである。DNNにおいて入力として与えられるベクトルは、基本的にすべて同次元である必要がある。対象のデータが画像である場合、一般的には入力を画像の各画素値を並べたベクトルとして表現するため、全データの画像サイズを統一する必要がある。しかしながら、基本的には移手段が徒歩であれば移動範囲が狭く、自動車であれば広がるなど、各セグメントによって移動範囲が異なるため、全範囲を再現しようとするセグメントごとに異なるサイズの画像が生成されてしまう。この問題を解決する方法の1つとして、セグメントごとに異なるサイズの画像を生成したのちに、拡大縮小により同サイズに揃える方法が考えられるが、各セグメントの縮尺を統一できなくなり、距離の情報が失われてしまう。2つ目の問題は、経緯度という高精度な単位で表現されたGPSログを画像化の際に、量子化により情報が失われてしまうことである。具体的には、測位点系列では表現できていた細かな動きをとらえきれなくなってしまう。たとえば、GPSによる測位点は移動していても、移動前後の2地点が同じ画素と対応してしまう可能性がある。画像サイズを大きくすれば1画素に対応する地理的範囲が小さくなるため、そのような動きをとらえることも可能だが、データ量が膨大になり現実的なりソースでの処理が難しくなる。さらに、全画素に対してGPS軌跡に関わる情報を持つ画素が少なくなるため、十分な学習をするためには大量のデータが必要となる恐れがある。

これらの問題に対して本研究では次のアプローチで軌跡画像を生成する。1つ目の問題に対しては、あらかじめ定義した緯度経度の範囲でセグメントから軌跡画像を切り出すことで、縮尺が同一の軌跡画像を生成するアプローチをとる。切り出す範囲外の情報は失われてしまうものの、我々は拡大縮小によるアプローチよりも、本アプローチが高い精度を示すことを予備実験で確認している。2つ目の問題に対しては、画像サイズを大きくすることで細かな動きを表現しようとするのではなく、画像サイズを小さくし

ながらも、滞在時間として各画素に情報を残すアプローチをとる。たとえばユーザが同じ画素に滞在している場合は、その滞在時間に応じて画素の値を大きくする。すなわち、グレースケール画像のように1つの画素に1つの値を持たせ、値が大きいほど可視化した際に明度が大きくなるようにデータを生成する。ただし、実際のグレースケール画像と異なり、大きな滞在時間も考慮するために画素値に上限は定めない。以上のアプローチによって、GPSログから移手段に特徴的な情報を保持した軌跡画像生成を実現する。

軌跡画像生成の概要を図3に、具体的な処理の流れをAlgorithm 1に示す。最初に入力データを定義する。入力として与えられたセグメント  $s$  におけるGPSログの  $N_s$  個の測位点系列を  $P_s = (p^{(i)})_{i=1}^{N_s}$  とする。 $i$  番目の測位点  $p^{(i)}$  は緯度  $p^{(i)}.lat$ 、経度  $p^{(i)}.lng$  および時刻  $p^{(i)}.t$  の情報を保持する。

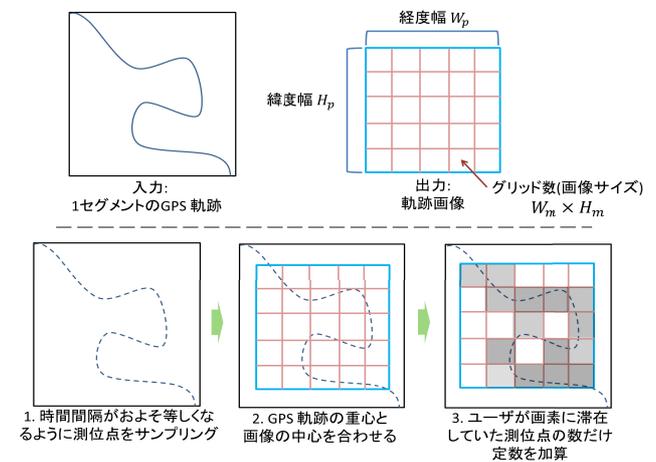


図3 軌跡画像生成の概要

Fig. 3 Overview of trajectory image generation.

#### Algorithm 1 EXTRACTIMAGEFROMGPSTRAJECTORY

**Input:**  $P_s = (p^{(i)})_{i=1}^{N_s}$ ,  $T$ ,  $W_p$ ,  $H_p$ ,  $W_m$ ,  $H_m$

**Output:**  $I_s$

**Initialize:**  $I_s \in \mathbb{R}^{W_m \times H_m} \leftarrow 0$

- 1:  $P'_s = \text{SAMPLINGPOINTS}(P_s, T)$
- 2:  $center_{lng} \leftarrow \frac{1}{|P'_s|} \sum_{j=1}^{|P'_s|} p^{(j)}.lng$
- 3:  $center_{lat} \leftarrow \frac{1}{|P'_s|} \sum_{j=1}^{|P'_s|} p^{(j)}.lat$
- 4:  $min_{lng} \leftarrow \min_j p^{(j)}.lng$
- 5:  $min_{lat} \leftarrow \min_j p^{(j)}.lat$
- 6:  $offset_x \leftarrow \lfloor \frac{W_m}{2} \rfloor - \lfloor (center_{lng} - min_{lng}) \frac{W_m}{W_p} \rfloor$
- 7:  $offset_y \leftarrow \lfloor \frac{H_m}{2} \rfloor - \lfloor (center_{lat} - min_{lat}) \frac{H_m}{H_p} \rfloor$
- 8: **FOR**  $j$  in 1 to  $|P'_s|$  **DO**
- 9:  $x \leftarrow \lfloor (p^{(j)}.lng - min_{lng}) \frac{W_m}{W_p} \rfloor + offset_x$
- 10:  $y \leftarrow \lfloor (p^{(j)}.lat - min_{lat}) \frac{H_m}{H_p} \rfloor + offset_y$
- 11: **IF**  $0 \leq x < W_m$  **AND**  $0 \leq y < H_m$  **THEN**
- 12:  $I_s(x, y) \leftarrow I_s(x, y) + 1$
- 13: **ENDIF**
- 14: **ENDFOR**

軌跡画像生成のためには、まず GPS ログの測位点系列の時間間隔を揃える処理を行う (ステップ 1)。各セグメントにおける測位点は、測位に使用される端末や電波状況の違いにより、すべて同じ時間間隔で測位されているとは限らない。時間間隔を揃えずにすべての測位点を画像にプロットしてしまうと、測位間隔が短いただけで同じ画素に長く滞在していると見なされてしまう。そこで、測位時刻  $p^{(i)}.t$  をもとに測位点系列  $P_s$  から時間  $T$  の間隔で測位点をサンプリングすることで、GPS の測位点系列の時間間隔を可能な限り統一する。つまり  $T$  に比較的大きな値を用いることで、広い時間間隔で測位されているセグメントに、全セグメントの測位点の時間間隔を揃えることができる。結果として、サンプリングして得られた測位点系列  $P'_s$  を用いて軌跡画像を生成する。

次に軌跡画像を生成する GPS ログの範囲を決定する (ステップ 2-7)。本手法ではサンプリングした GPS ログ  $P'_s$  の重心が画像の中心と一致するように、すべてのセグメントに対して同様の条件で軌跡画像を切り出す。切り出す範囲は緯度経度座標空間において経度幅  $W_p$ 、緯度幅  $H_p$  の矩形領域とする。この際に矩形領域は  $W_m \times H_m$  のグリッドに分割され、各グリッドが画像の各画素と対応する。すなわち得られる画像サイズは  $W_m \times H_m$  となる。Algorithm 1 においては、GPS ログの重心と最も南西にある測位点をもとに、画像の中心と GPS ログの重心を揃えるための補正値を算出する。

最後に全測位点について、矩形領域内に測位点が存在している場合、画像空間上に測位点をプロットする (ステップ 8-14)。実際は Algorithm 1 に示すように、前処理で算出した補正値をもとに、測位点と対応する画像空間上の座標を求め、指定された画像サイズ内に収まれば、測位点をプロットする。プロットの際は各画素における滞在時間を表現するために、定数を加算していく。以上の処理の結果、最終的に軌跡画像  $I_s \in \mathbb{R}^{W_m \times H_m}$  が得られる。

測位点系列の時間間隔を揃える方法として、提案手法ではサンプリング処理を行っているが、別の方法も考えられる。たとえば、ある測位点における滞在時間を 1 つ前の測位点との時間間隔と仮定し、この値を直接画素値に代入する方法が考えられる。または 2 つの測位点を結ぶ線分を、1 秒など一定時間おきになるように細分して、中間点に仮想的な測位点を割り当てることで軌跡を補間することも可能である。これらの方法は、サンプリングによる方法と異なり測位された情報を失うことはないが、それぞれ次のような問題がある。前者の方法に関しては、GPS の受信状況などによって一時的に測位間隔があいた場合など離れた 2 点の測位点が存在する場合、測位時間間隔を滞在時間としてしまうと、その 2 点間を移動する時間が考慮されないという問題が生じる。後者の方法に関しては、1 秒など一定時間間隔で点を補うことで、測位点が一定速度で移動

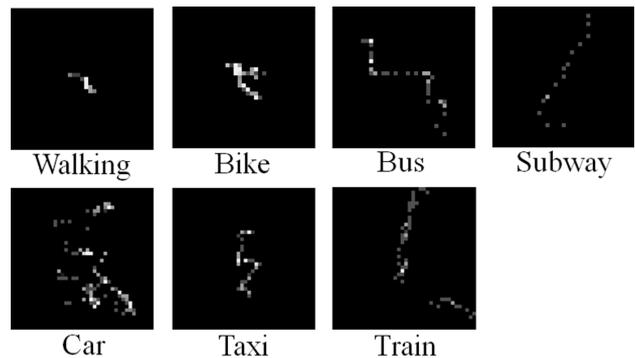


図 4 実データから生成した軌跡画像の例。画素の色が明るいほどユーザーがその場所に長く滞在していることを表す。同じ縮尺で切り取ることで距離の情報を保持し、さらに時間に関する情報を画素値で表現できていることが分かる

Fig. 4 Examples of trajectory images extracted from real data. Brighter color means longer stay time. These trajectory images can store distance information by clipping trajectories in same range and represent time information as stay time in each pixel value.

しているという情報を与えることになる。しかし、実際は測位点の間を等速で移動しているとは限らないため、補間により現実とは異なる情報を与えてしまい、特徴の抽出に悪い影響を与えてしまう可能性がある。このように本来の GPS ログとはかい離した情報を付与してしまう恐れがあるため、今回はサンプリングによって測位点の時間間隔を揃えると同時に、実際に測位された点の情報のみを利用して軌跡画像を生成している。

図 4 は実データの GPS ログから生成された軌跡画像の一例である。軌跡画像をグレースケール画像として可視化するために、軌跡画像の各画素の値が  $[0, 255]$  に収まるようにスケーリングし、256 階調になるよう量子化を行っている。ユーザーが同じ地点に滞在する時間が長いほど対応する画素が明るくなっている。同じ縮尺で切り取ることで距離の情報を保持し、さらに時間に関する情報を画素値で表現できていることが分かる。この例では歩行 (walking) は基本的に他の移動手段と比べ、移動距離も小さく同じ地点に滞在しているため、画像の中心付近が明るくなる傾向にある。一方でバス (bus) や地下鉄 (subway) は直線的な移動が多く広範囲にわたった軌跡であることが分かる。単純に移動距離のみで移動手段を判別できる場合もあれば、速度や加速度情報を用いても適切に判別できない場合もある。本研究では軌跡画像に対して deep learning を適用し、判別に有効な特徴の自動獲得を行う。

### 3.2 Deep learning による軌跡画像からの特徴抽出

GPS ログから生成された軌跡画像を入力として DNN により特徴を学習する。本研究では DNN の pre-training として SDA [22] を用いる。本節では主に SDA の要素技術である Autoencoder (AE) [1] と Denoising Autoencoder

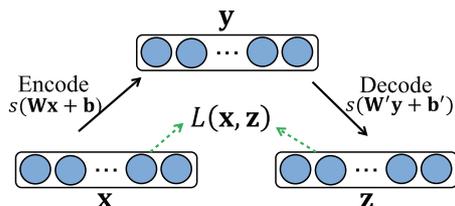


図 5 AE の構造. 入力ベクトル  $\mathbf{x}$  をエンコードおよびデコードして得られた  $\mathbf{z}$  が  $\mathbf{x}$  を再現できるように, 再現誤差  $L(\mathbf{x}, \mathbf{z})$  を最小化するようにパラメータを学習する

Fig. 5 Structure of AE. Parameters are learned so that reconstruction error  $L(\mathbf{x}, \mathbf{z})$  between input vector  $\mathbf{x}$  and encoded and decoded vector  $\mathbf{z}$  is minimized.

(DAE) [21], および SDA について簡潔に説明する.

AE は入力として与えられたベクトル  $\mathbf{x}$  をエンコードすると得られる中間層のベクトル  $\mathbf{y}$  を, さらにデコードした際に入力  $\mathbf{x}$  を復元できる恒等写像を学習する手法である (図 5 参照). エンコードは次式のとおり入力ベクトル  $\mathbf{x}$  を非線形変換する:

$$\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b}). \quad (1)$$

ここで  $\mathbf{W} \in \mathbb{R}^{n_h \times n_x}$  は  $n_x$  次元の入力層から  $n_h$  次元の中間層への変換を行う重み行列であり,  $\mathbf{b} \in \mathbb{R}^{n_h}$  はバイアス項である.  $s(\cdot)$  は活性化関数であり, たとえば sigmoid 関数や tanh 関数を用いる. また, デコードは次式に示すとおりエンコードによって得られた中間層のベクトル  $\mathbf{y}$  をさらに非線形変換する:

$$\mathbf{z} = s(\mathbf{W}'\mathbf{y} + \mathbf{b}'). \quad (2)$$

ここで  $\mathbf{W}' \in \mathbb{R}^{n_x \times n_h}$  は  $n_h$  次元の中間層から  $n_x$  次元の出力層への変換を行う重み行列であり,  $\mathbf{b}' \in \mathbb{R}^{n_x}$  はバイアス項である. 重み行列は tied weights と呼ばれる  $\mathbf{W}' = \mathbf{W}^T$  という制約をおく場合もあり, これによりパラメータの自由度を減らしモデルの過学習を防ぐ効果があると考えられる. AE は入力層に与えられたベクトル  $\mathbf{x}$  と出力  $\mathbf{z}$  が同じになるように, 再現誤差  $L$  を最小化する重み行列とバイアス項を求めることで学習を行う. 再現誤差は  $P(\mathbf{x}|\mathbf{z})$  に対して仮定する分布によって異なり, たとえばガウス分布を仮定する場合は二乗誤差として定義される:

$$L(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|^2. \quad (3)$$

また,  $P(\mathbf{x}|\mathbf{z})$  をベルヌーイ分布や二項分布と仮定する場合は, クロスエントロピー誤差が用いられる. すなわち入力ベクトル  $\mathbf{x}$  の変域が  $[0, 1]^{n_x}$  である場合はクロスエントロピー誤差が用いられ,  $\mathbf{x}$  の変域が特に限定されない場合は二乗誤差が用いられることが一般的である. 本研究においては, 変域が特に限定されない実数値からなる軌跡画像が入力であるため, 再現誤差として二乗誤差を利用する. 再現誤差に基づくパラメータの最適化には, 確率的勾配降下 (SGD) 法や L-BFGS 法が用いられることが多い.

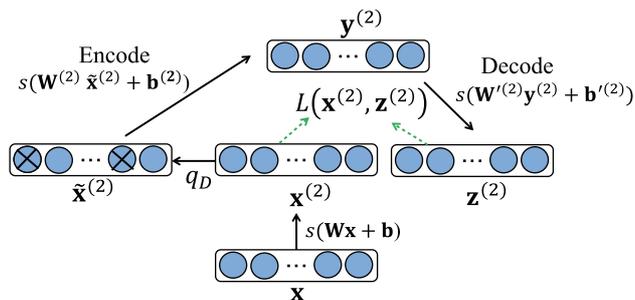


図 6 SDA の構造. DAE によって得られた中間層の出力を再度別の DAE の入力として用い学習する. これを繰り返すことで中間層を多段に積み上げていく.  $\mathbf{x}^{(2)}$  は 2 層目の DAE の入力を表し, 添字は層の番号を示している. 具体的には 1 番目の DAE の中間層の出力を, 新たに用意された DAE の入力と見なして重み行列  $\mathbf{W}^{(2)}$  とバイアス項  $\mathbf{b}^{(2)}$  を学習する. その結果得られた  $\mathbf{y}^{(2)}$  を 3 層目の DAE の入力  $\mathbf{x}^{(3)}$  として用いる

Fig. 6 Structure of SDA. Intermediate representations at deepest layer at current stack of DAEs are used as input for new DAE when new layer is added. In this example, first intermediate representation  $\mathbf{x}^{(2)}$  at second layer is used as input of newly arranged DAE, and weighting matrix  $\mathbf{W}^{(2)}$  and bias term  $\mathbf{b}^{(2)}$  are learned (where upper indices indicate layer number). Then  $\mathbf{y}^{(2)}$  is used as input of DAE at third layer (i.e.,  $\mathbf{x}^{(3)}$ ).

DAE はノイズを加えた入力ベクトルに AE と同様のエンコードとデコードを行い, ノイズが取り除かれた元の入力ベクトルを再構築する写像を学習する. これにより AE の汎化性能を向上させることができると考えられている. 具体的には, 入力ベクトル  $\mathbf{x}$  における各要素の一部にノイズを加える分布  $q_D(\tilde{\mathbf{x}}|\mathbf{x})$  を仮定し, 分布に基づいてノイズの付与された入力ベクトル  $\tilde{\mathbf{x}}$  をサンプリングする.  $\tilde{\mathbf{x}}$  を用いて AE と同様の処理で得られた  $\mathbf{z}$  が, 元の入力ベクトル  $\mathbf{x}$  に近づくように式 (3) と同様に誤差関数を最小化する.

SDA は DAE を多段に構成した手法であり, DNN の初期化として用いられる (図 6 参照). SDA は DAE によって得られた中間層の出力をさらに別の DAE の入力で見なして学習を繰り返すことで, 複数の中間層を積み上げていく. この際 DAE の学習に用いる入力には, 前の段階で積み上げられた DNN を用いて, データから得られた最も深い中間層のベクトルを用いる. 最終的にラベル付きデータから得られた最も深い中間表現を教師あり学習の入力として用いることができる.

本研究では SDA の入力として, 軌跡画像の行列  $\mathbf{I}_s$  を  $W_m \times H_m$  次元のベクトルに変換したものをを用いる. ノイズを付与する分布  $q_D(\tilde{\mathbf{x}}|\mathbf{x})$  についてはベルヌーイ分布を仮定し, ノイズを発生させる割合 (corruption rate) に基づき, 入力ベクトル  $\mathbf{x}$  の要素の一部が 0 となるような masking noise [22] を発生させる. SDA の重み行列やバイアス項の最適化には, SGD 法と比較した収束性の速さや, 分類問題における精度向上が報告されている minibatch L-BFGS 法

を用いる [10]。また、重み行列については tied-weights を用いる。SDA によって教師なし学習を行った後は、ラベル付きデータをもとに fine-tuning によって DNN の全体のパラメータを調節する。fine-tuning の際は、ラベルの種類数のニューロンからなる出力層を追加し、セグメントに付与された教師ラベルを表すバイナリベクトルとの二乗誤差をもとに SGD 法によりパラメータを更新する。各ニューロンに対する活性化関数にはすべて sigmoid 関数を用いた。以上の処理によって得られた出力層に最も近い中間層の出力を新たな特徴として用いる。これと従来の feature engineering で得られた特徴ベクトルを結合し、結合した特徴ベクトルを用いて教師あり学習を行い分類器を構築する。

#### 4. 評価

本章では評価実験を通じて、提案手法により得られた特徴を用いて、教師あり学習による移動手段推定を行った際に、従来のアプローチに比べて高精度な推定が達成可能であることを検証する。まずデータセットについて述べたのちに、評価実験の結果を述べる。

データセット. 本実験では、Microsoft Research によって公開されている GeoLife データセット [25], [26], [27]<sup>\*4</sup> を利用した。本データセットの GPS ログは基本的に 1-3 秒ごとに測位されており、測位点は緯度経度と時刻の情報を持つ。データセットには、2007 年 4 月から 2012 年 8 月にかけて 182 ユーザの GPS ログが含まれており、そのうち 69 ユーザが一部の GPS ログに対して walking や car など 11 種類の移動手段のアノテーションを行っている。69 ユーザのうち、アノテーションの数が 10 未満の極端に少ないユーザはデータセットから除外し、計 54 ユーザのデータを本実験に用いることとした。アノテーションには移動手段を表すラベルと移動手段の開始時刻と終了時刻が記載されている。本実験では、開始時刻と終了時刻の区間を 1 つのセグメントとして、各セグメントにラベルの付与を行い、これらを実験に用いた。データセットは全 11 種類のラベルを含むが、うち 4 種類のラベルは数が非常に少ないため、本実験では 7 種類の移動手段 (walking, bus, car, bike, taxi, subway, train) を評価対象とした。結果として、合計で 9,043 セグメントが得られた。データセットはユーザ単位で訓練データとテストデータに分割し、5 分割交差検定 (5-fold Cross Validation; CV) によって評価を行った。評価指標にはテストデータのうちの何割のセグメントに対して、正確に移動手段を推定できたかを表す正解率 (Accuracy) を用いた。

##### 4.1 表現学習による軌跡画像からの特徴抽出の有効性評価 従来の移動手段推定に用いられていた feature engineer-

ing による特徴と、提案手法の表現学習による特徴を用いて教師あり学習を行い移動手段の推定精度を比較することで、提案手法の有効性を検証する。さらに、画像認識の分野で一般的に用いられている特徴表現である Bag of Visual Words (BoVW) との比較により、表現学習を利用することの有効性を検証する。具体的には下記の特徴を用いて比較する。

- Basic Features (BF) [25]: 移動距離, 速度や加速度など計 10 次元の基本特徴.
- BF+Advanced Features (AF) [27]: BF に加えて VCR, SR, HCR を用いる計 13 次元の発展的特徴.
- BoVW: Dense-SIFT [20] を用いて軌跡画像から得られた特徴.
- SDA: SDA を用いて軌跡画像から得られた特徴 (提案手法 1).
- BF+AF+SDA: Feature engineering に基づく BF および AF と、表現学習に基づく SDA を結合した特徴 (提案手法 2).

BoVW の代表的な方法として SIFT [14] がある。SIFT は画像において視覚的に顕著な点 (key point) から、照明変化や幾何変化に頑健な 128 次元の局所特徴を計算する。この際、訓練データから得られた複数の局所特徴を量子化することでコードブックを生成する。得られたコードブックを用いて、訓練データやテストデータの局所特徴からヒストグラムを生成し、これを特徴として利用する。本実験の入力となる軌跡画像においては視覚的に顕著な画素が少なく、key point をまったく検出できない画像が多く存在した。そのため今回はすべての画素において特徴を計算する Dense-SIFT [20] を利用した。またヒストグラム生成における bin の数は任意のパラメータであるが、本実験では {10, 50, 100, 200, 500, 1,000} の中で最も高い推定精度を得られる 500 を用いた。

これらの特徴をもとに、マルチクラス分類が可能な機械学習手法を用いて教師あり学習を行う。BF や BF+AF については、既存研究 [25], [27] において、最も高い精度を示したと報告されている Decision Tree (DT) を用いた。BoVW については、Logistic Regression (LR), Support Vector Machine (SVM), DT のうち最も高い精度を示した SVM を用いた。SDA, BF+AF+SDA については最も高い精度を示した LR を用いた。

DT のパラメータ探索については、枝分割の方法 (Gini 係数またはエントロピー) と分類に利用する最大特徴 {0.1, 0.2, ..., 1.0} から、訓練データに対する 5-fold CV<sup>\*5</sup> を用いてグリッドサーチを行い、正解率が最大になるパラメー

<sup>\*4</sup> <http://research.microsoft.com/en-us/downloads/b16d359d-164-469e-9fd4-daa38f2b2e13/>

<sup>\*5</sup> 本実験では、評価のための 5-fold CV の各試行において、パラメータ探索のために訓練データに対して 5-fold CV を用いており、両者が異なることに注意されたい。これは nested CV と呼ばれる方法である。

タを選択した。SVMについてはRBFカーネルを用い、トレードオフパラメータを $\{0.01, 0.1, 1, 10, 100\}$ から、カーネル係数を $\{0.001, 0.01, 0.1, 1, 10\}$ からDTと同様の方法で探索した。軌跡画像生成およびDNNのパラメータについても基本的にグリッドサーチを行うが、探索数を減らすために次のように探索方法を簡易化した。まずパラメータを固定したDNNを用いて、軌跡画像生成のパラメータについてグリッドサーチを行う。グリッドサーチの結果正解率が最大になる軌跡画像を用いて、DNNのパラメータを探索する。軌跡画像生成については、測位点のサンプリング間隔 $T$ を $\{10, 30, 60, 120\}$ 秒から、経度幅 $W_p$ および経度幅 $H_p$ を $\{0.01, 0.05, 0.1, 0.2\}$ 度<sup>\*6</sup>から、画像サイズ $W_m \times H_m$ を $\{20 \times 20, 25 \times 25, 30 \times 30, 35 \times 35, 40 \times 40, 50 \times 50\}$ から探索した。DNNについては、中間層の数を $\{1, 2, \dots, 5\}$ から、各層のニューロンの数を $\{10, 50, 100, 200\}$ から、corruption rateを $\{0, 0.2, 0.4, 0.6, 0.8\}$ から探索した。またfine-tuningのSGD法による最適化のパラメータとして、学習率は0.1に固定し、学習回数を $\{1, 2, \dots, 15\}$ から探索した。ここで1回の学習とは、SGD法により全データを1回用いパラメータを更新することを指す。

#### 4.1.1 結果

図7に結果を示す。グラフの値は5-fold CVによる正解率の平均値を表し、エラーバーは標準偏差を表す。まず、基本的な特徴を用いたBFに対して、より複雑な特徴を追加したBF+AFを用いることで1.3%ポイントの精度の向上を確認できる。BoVWでは軌跡画像のみを用いているにもかかわらず、60.2%の精度を達成できているが、feature engineeringによるBFやBF+AFの正解率には届いていない。これに対して、SDAはBoVWと比べ高い精度を示しており、AFと比べても1.5%ポイントの精度の向上を示している。これは、表現学習により移動手段の推定に有効な

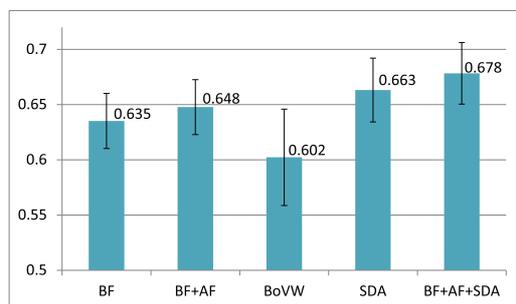


図7 各手法による移動手段推定の正解率。正解率は0から1の範囲をとるが、違いを分かりやすくするために0.5から0.7の範囲でグラフを表示している

Fig. 7 Comparisons of performance of transportation mode estimation on each feature extraction method. All accuracies are in the 0 to 1 range, but they are shown in the 0.4 to 0.7 range for high visibility.

\*6 緯度経度が0.01度につき1,000メートル前後の距離を表す。

特徴を抽出できたためと考えられる。さらに、表現学習によるSDAにfeature engineeringによるBFとAFを結合したBF+AF+SDAを用いることで、SDAよりも1.5%ポイントのさらなる精度の向上が確認できる。これはBFからBF+AEへの精度の上昇量に匹敵する値である。このような結果から、人手で設計することが難しい特徴を、表現学習によって効果的に抽出できたといえる。

さらに図8を通して、推定精度を移動手段ごとに詳細に分析する。ここで、(a)は既存手法で最も精度の高いBF+AFによる推定精度の混同行列を表し、(b)は提案手法で最も精度の高いBF+AF+SDAによる混同行列を表す。混同行列の水平軸は予測結果、垂直軸は正解アノテーションに該当し、行列の各要素には5-fold CVの各試行における判定結果の和を行ごとに正規化した値が示されている。行列の対角成分を見ると、提案手法は既存手法と比べ7種類の移動手段のうちbike, subway, trainおよびwalkingの4種類において精度の向上が見られ、残りのbus, carおよびtaxiの3種類は既存手法と同程度の精度を確認できる。特にwalking, subwayおよびtrainの精度向上は顕著であり、これらの移動手段を区別するために有効な特徴が獲得できたといえる。しかし、bus, carおよびtaxiは乗り物としては同じ自動車であるためか、既存手法と同様に提案手法で区別することも容易ではなく、他の移動手段と比べても推定精度が低い。一方で、混同行列の列成分に着目すると、これら3つの移動手段であると推定されたもののうちbus, carやtaxi以外の移動手段と誤推定されるものが減少している傾向が見られる。特にsubwayやtrainとアノテーションされたもののうち、busと誤推定された割合は、提案手法が既存手法を大きく下回っている。また、提案手法によるcarやtaxiの推定精度は既存手法から大きく変化していないにもかかわらず、carやtaxiとアノテーションされたものがbusと推定された割合は提案手法が既存手法を上回っている。これらの結果から、bus, carやtaxiのような自動車に分類される移動手段とその他の移動手段を区別するために有効な特徴を獲得できていると考えられる。

前述のとおりbus, carやtaxiは乗り物としては同じ自動車であるため区別は容易ではないが、このような細かな移動手段推定を実現することで、都市計画やサービスへの応用の幅を広げることができると考えられる。これらの推定精度を向上させる方法として、軌跡画像のサイズを大きくすると同時に、訓練データを増やすことが考えられる。軌跡画像の画像サイズを大きくすることで画像の解像度が増し、ユーザの移動軌跡をより多くの画素で表現できる。これにより、より細かなユーザの動きをとらえられることから、deep learningによって獲得できる特徴の幅も広がり、多様な移動手段をより正確に区別できる可能性がある。しかしながら、今回の実験におけるパラメータ探索範囲の最大値である $50 \times 50$ よりも画像サイズを大きくしていくと、

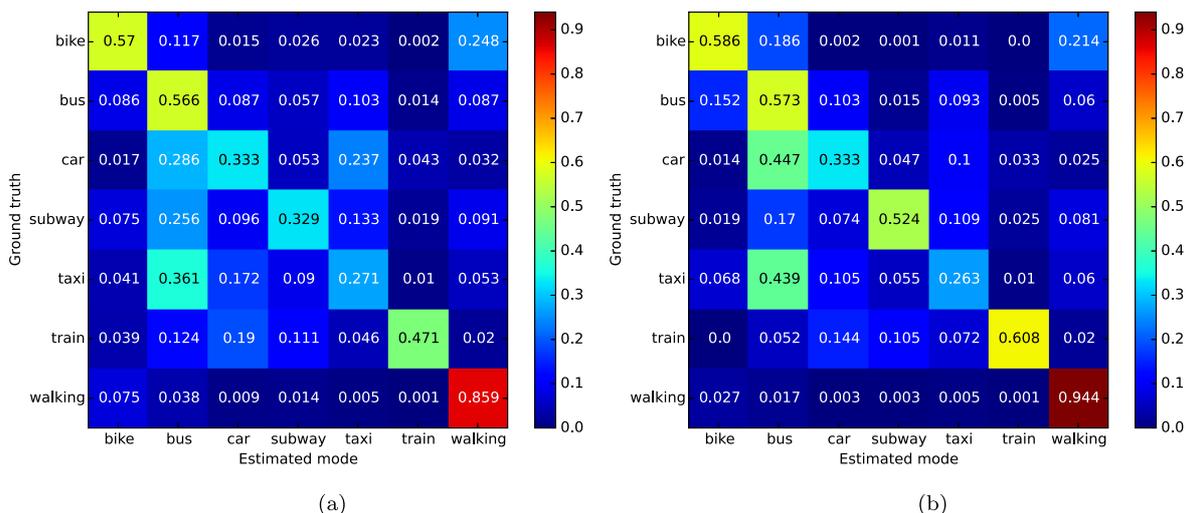


図 8 移動手段ごとの正解率の比較. (a) 既存手法 (BF+AF) によって得られた混同行列. (b) 提案手法 (BF+AF+SDA) によって得られた混同行列

Fig. 8 Comparisons of performance on each transportation mode. (a) confusion matrix of previous method (BF+AF). (b) confusion matrix of proposed method (BF+AF+SDA).

徐々に推定精度が低下していく結果となった. これは, 画像サイズを大きくすることで細かな動きをとらえられる一方で, 画素数の増大にともない入力ベクトルの次元が大きくなり, DNN が訓練データにオーバフィッティングしてしまったためであると考えられる\*7. 訓練データ数を増やせばこの問題は解決するかもしれないが, 移動軌跡へのアノテーションはコストがかかる. そこでこのような問題に対する 1 つの解決方法として, ラベルなしデータも利用できる半教師あり学習の枠組みで DNN を学習するアプローチが考えられる. ラベルなしデータであればアノテーションコストがかからず, 比較的容易にデータを獲得することができる. Pre-training の段階では正解ラベルを必要としないため, ラベルなしデータとラベル付きデータの両方を用いて学習し, fine-tuning のみラベル付きデータのみを利用してモデルを更新することで, モデルの汎化能力を向上させることが可能になる. このような方法により, より詳細な移動手段の推定を実現できる可能性があるため, 今後さらなる検討を行ってきたい.

#### 4.2 GPS ログに対する表現学習の評価

本節では, 画像化した GPS ログに対して適切に表現学習を行えているかを, 学習した DNN における各ニューロンを可視化することで検証する.

図 9 に示す上の 2 つの画像は, 軌跡画像から学習した DNN の各中間層におけるニューロンの活性化状態を可視化し並べた結果である. 移動範囲や移動間隔の異なる軌跡

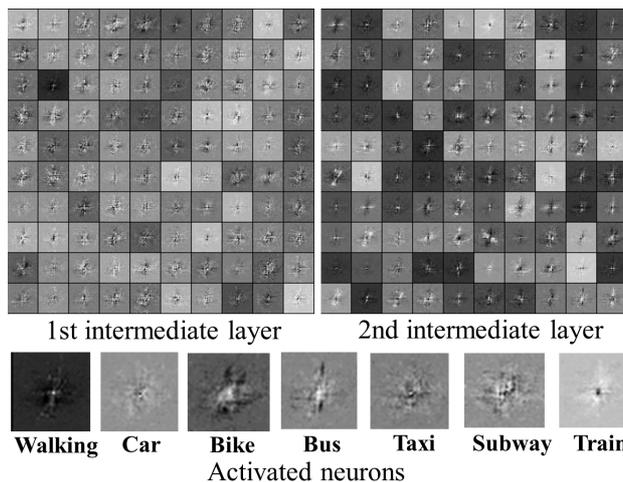


図 9 中間層を 3 層, 100 ニューロンで学習した際の, 各中間層のニューロン (重み行列) の可視化結果 (上), および 1 層目または 2 層目の中間層において各移動手段の GPS ログに強く反応するニューロンの可視化結果 (下)

Fig. 9 Visualization results on state of neurons (weighting matrices) on learned DNN when DNN is learned with three intermediate layers and 100 neurons on each layer. Two upper images show states of neurons on 1st and 2nd intermediate layers, and seven lower images show states of neurons strongly responding to data with labels of each transportation mode.

や, 直線的な軌跡など, 各中間層の重みが軌跡画像から特徴的な部分を抽出するフィルタとして働いていることがうかがえる. また中間層が出力層に近づくにつれて, 特徴がより抽象化されていると考えられる.

図 9 における下の 7 つの画像は, 各移動手段ラベルの付与されたデータに対して, 強く反応するニューロンの活性

\*7 たとえば画像サイズが  $50 \times 50$  であれば 2,500 次元,  $100 \times 100$  であれば 10,000 次元と, 入力ベクトルの次元は画像の幅に対して加速度的に増加する.

状態を可視化し拡大したものである。可視化によって学習した重みの特徴をすべて理解するのは難しいが、移動範囲を見ても歩行 (walking), 自転車 (bike), バス (bus) の順に広がっていることが分かる。また、バスと自動車 (car) を比べるとバスが暗い領域をより多く含むが、これはバスが決められた道しか走らないためであると予想される。このように移動手段ごとに反応するニューロンが異なり、移動手段の区別に有効な特徴を抽出していると考えられる。

### 4.3 DNN のパラメータごとの推定精度の比較

表 1 は DNN における中間層の数, ニューロンの数および corruption rate を変更した際の推定精度の違いを表す。太字は精度が最も高いものを表している。中間層の数については 3 層が、ニューロンの数については 100 個が最も高い精度を示す結果となった。ここから得られる知見は、中間層の数やニューロンの数を多くしすぎると、モデルの表現力が高くなりすぎて、訓練データに対して過学習していることである。逆にこれらの数が少ない場合、十分に特徴を表現しきれず、精度も低くなってしまふ。また、corruption rate については 0.4 が最も高い精度を示している。ある程度入力信号にノイズを加えることで、モデルの汎化性能が向上し、精度が向上したと考えられる。

表 2 は DNN の学習時に前処理として、SDA により pre-training を行い得られたパラメータを fine-tuning の初期値として用いた場合と、pre-training を行わずにランダムに初期化したパラメータを fine-tuning の初期値として

表 1 DNN のパラメータによる推定精度の違い

Table 1 Results with various DNN parameters.

N-layer	N-neuron	Corruption rate	Accuracy
1-layer	100-neuron	0.2	0.633 ± 0.068
2-layer	100-neuron	0.2	0.652 ± 0.059
3-layer	100-neuron	0.2	0.683 ± 0.059
4-layer	100-neuron	0.2	0.677 ± 0.058
5-layer	100-neuron	0.2	0.680 ± 0.065
3-layer	10-neuron	0.2	0.607 ± 0.091
3-layer	50-neuron	0.2	0.657 ± 0.067
3-layer	100-neuron	0.2	0.683 ± 0.059
3-layer	200-neuron	0.2	0.671 ± 0.069
3-layer	100-neuron	0.0	0.669 ± 0.057
3-layer	100-neuron	0.2	0.683 ± 0.059
<b>3-layer</b>	<b>100-neuron</b>	<b>0.4</b>	<b>0.687 ± 0.051</b>
3-layer	100-neuron	0.6	0.683 ± 0.053
3-layer	100-neuron	0.8	0.673 ± 0.052

表 2 Pre-training の有無による推定精度の違い

Table 2 Results w/ and w/o pre-training.

Alg.	Accuracy
fine-tuning w/o pre-training	0.624 ± 0.033
<b>fine-tuning w/ pre-training</b>	<b>0.678 ± 0.028</b>

用いた場合の精度を示している。Pre-training を行わない場合、大きく精度が低下している。これにより、SDA が表現学習の観点で重要な役割を果たしていることが確認できる。

## 5. おわりに

本稿では、移動手段推定のために GPS ログから軌跡画像を生成し、deep learning の枠組みによって高次の特徴を抽出する手法を提案した。我々の知る限り、移動手段推定タスクにおける deep learning の適用は初めての試みである。実データを用いた評価実験では、軌跡画像から deep learning を利用することで得られた特徴を用いることで、従来の feature engineering と同等の推定精度を達成できた。さらに移動距離、速度、加速度、停止率や方向変換率を含む従来の特徴に、提案手法によって抽出された特徴を組み合わせて学習することで、推定精度が向上することを確認した。この結果から、軌跡を画像としてとらえて、適切な範囲に対して適切なフィルタを用意することができる deep learning を利用することで、人手による設計が困難な特徴を抽出することができたと考えられる。

現状の問題としては、GPS ログの画像化の際に、縮尺を維持するために緯度経度幅を固定しているため、範囲外の情報が失われていることがあげられる。また、時間的な情報として滞在時間の情報は考慮されているが、系列的な情報は考慮されていない。今後の課題としては、このような情報を有効活用した特徴抽出や、前述の半教師あり学習によるアプローチの検討があげられる。また、本稿で提案した GPS ログからの特徴抽出の適用先は、移動手段推定の問題に限定されないと考える。今後は他のタスクへの応用も検討していきたい。

## 参考文献

- [1] Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H.: Greedy layer-wise training of deep networks, *Proc. NIPS '06*, pp.153-160 (2006).
- [2] Bengio, Y.: Learning deep architectures for AI, *FTML*, Vol.2, No.1, pp.1-127 (2009).
- [3] Cho, E., Myers, S.A. and Leskovec, J.: Friendship and mobility: User movement in location-based social networks, *Proc. KDD '11*, pp.1082-1090 (2011).
- [4] Collobert, R. and Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning, *Proc. ICML '08*, pp.160-167 (2008).
- [5] Dahl, G.E., Yu, D., Deng, L. and Acero A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition, *TASLP*, Vol.20, No.1, pp.30-42 (2012).
- [6] Ermes, M., Parkka, J., Mantyjarvi, J. and Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions, *IEEE Trans. Inform. Tech. Biomed.*, Vol.12, No.1, pp.20-26 (2006).
- [7] Hinton, G.E. and Salakhutdinov, R.: Reducing the di-

mensionality of data with neural networks, *Science*, Vol.313, No.5786, pp.504-507 (2006).

[8] Hinton, G.E., Osindero, S. and Teh, Y.: A fast learning algorithm for deep belief nets, *Neural Computation*, Vol.18, No.7, pp.1527-1554 (2006).

[9] Krizhevsky, A., Sutskever, I. and Hinton, G.: Imagenet classification with deep convolutional neural networks, *Proc. NIPS '12*, pp.1106-1114 (2012).

[10] Le, Q.V., Ngiam, J., Coates, A., Lahiri, A., Prochnow, B. and Ng, A.Y.: On Optimization Methods for Deep Learning, *Proc. ICML '11* (2011).

[11] Le, Q.V., Ranzato, M., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J. and Ng, A.Y.: Building high-level features using large scale unsupervised learning, *Proc. ICML '12* (2012).

[12] Liao, L., Fox, D. and Kautz, H.: Learning and Inferring Transportation Routines, *Proc. AAAI '04*, pp.348-353 (2004).

[13] Liao, L., Fox, D. and Kautz, H.: Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields, *Int. J. Rob. Res.*, Vol.26, No.1, pp.119-134 (2007).

[14] Lowe, D.: Distinctive Image Features from Scale-invariant Keypoints, *Int. J. Comput. Vision*, Vol.60, No.2, pp.91-110 (2004).

[15] Nishida, K., Toda, H., Kurashima, T. and Suhara, Y.: Probabilistic Identification of Visited Point-of-Interest for Personalized Automatic Check-in, *Proc. UbiComp '14* (2014).

[16] Parkka, J., Ermes, M., Korpippa, P., Mantjarvi, J. and Peltola, J.: Activity classification using realistic data from wearable sensors, *IEEE Trans. Inform. Tech. Biomed.*, Vol.10, No.1, pp.119-128 (2006).

[17] Patterson, D., Liao, L., Fox, D. and Kautz, H.: Inferring high-level behavior from low-level sensors, *Proc. UbiComp '03*, pp.73-89 (2003).

[18] Ranzato, M. and Szummer, M.: Semi-supervised learning of compact document representations with deep networks, *Proc. ICML '08*, pp.792-799 (2008).

[19] Salakhutdinov, R. and Hinton, G.E.: Deep Boltzmann machines, *Proc. AISTATS '09*, pp.448-455 (2009).

[20] Vedaldi, A. and Fulkerson, B.: Vlfeat: An open and portable library of computer vision algorithms, *Proc. MM '10*, pp.1469-1472 (2010).

[21] Vincent, P., Larochelle, H., Bengio, Y. and Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders, *Proc. ICML '08*, pp.1096-1103 (2008).

[22] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y. and Manzagol, P.-A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, *JMLR*, Vol.11, pp.3371-3408 (2010).

[23] Xiao, X., Zheng, Y., Luo, Q. and Xie, X.: Finding similar users using category-based location history, *Proc. SIGSPATIAL '10*, pp.442-445 (2010).

[24] Yuan, J., Zheng, Y. and Xie, X.: Discovering regions of different functions in a city using human mobility and POIs, *Proc. KDD '12*, pp.186-194 (2012).

[25] Zheng, Y., Liu, L., Wang, L. and Xie, X.: Learning transportation mode from raw GPS data for geographic applications on the web, *Proc. WWW '08*, pp.247-256 (2008).

[26] Zheng, Y., Li, Q., Chen, Y., Xie, X. and Ma, W.-Y.: Understanding Mobility Based on GPS Data, *Proc. UbiComp '08*, pp.312-321 (2008).

[27] Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y.: Understanding transportation modes based on GPS data for web applications, *ACM Trans. Web*, Vol.4, No.1, pp.1:1-1:36 (2010).



遠藤 結城 (正会員)

2012年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士前期課程修了。同年日本電信電話株式会社に入社。以来、データマイニングの研究開発に従事。



戸田 浩之 (正会員)

1997年名古屋大学工学部材料プロセス工学科卒業。1999年同大学大学院工学研究科材料プロセス工学専攻博士課程前期課程修了。同年日本電信電話株式会社に入社。以来、情報検索、データマイニングの研究開発に従事。

2007年筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻博士後期課程修了。博士(工学)。ACM, 電子情報通信学会, 日本データベース学会各会員。



小池 義昌

日本電信電話株式会社サービスエボリューション研究所所属。1989年東北大学大学院材料化学専攻博士前期課程修了後、日本電信電話株式会社に入社。以来、パターン認識の研究、遠隔教育システムの研究開発、地域情報検索サービスの研究に従事。

索サービスの研究に従事。

(担当編集委員 北山 大輔)