修士論文

少量のラベルデータを用いた実践的な交通移動モード推定システムに関する研究

(Study on a Practical System for Transportation Mode Inference Using a Small Amount of Labeled Data)

芳賀 宣仁

(48-146436)

東京大学大学院 情報理工学系研究科 電子情報学専攻

指導教員　　安達 淳 教授

平成 28 年 2 月 4 日提出

概要

スマートフォンに代表される Global Positioning System 機能搭載機器の普及に従い, 近年では大量の位置情報を容易かつ安価に取得可能となっている. 位置情報に関する研究の一分野に, 人の交通移動モード推定がある. 交通移動モードとは車・自転車・徒歩などの移動形態のことである. 交通移動モード推定とは, 位置情報から得られる移動軌跡上の任意の位置における交通移動モードを推定することである.

既存の交通移動モード推定手法は, 教師あり学習手法を用いるため, 学習に用いる交通移動モードのラベル付き位置情報が必要となる. しかし, 交通移動モードのラベル付き位置情報の作成には, 大きな時間的・金銭的コストが掛かる. したがって, なるべく少ない量のラベル付き位置情報のみで推定することは, ラベル付け作業のコスト削減に対し有効となり重要である.

本研究の目的は, ラベル付き位置情報の量が少ない場合でも, ラベル付き位置情報が十分にある場合と同等の精度で交通移動モード推定を行うことである. その目的を達成する手段として, 半教師あり学習手法による推定を軸とした研究を行った. 本研究では, まず半教師あり学習手法を交通移動モード推定に用いた既存研究がないため, 4 つの既存の半教師あり学習手法を交通移動モード推定に適用し, どの手法が交通移動モード推定に有効であるかを実験で確認した. 実験の結果, Label propagation が交通移動モード推定に最も有効であることが確認された. ついで, Label propagation を拡張した交通移動モード推定手法を提案し, 提案手法を用いた交通移動モード推定システムを構築した.

提案手法の有効性を, 中国北京における実際の位置情報を用いて評価した. 実験の結果, 提案手法はラベル付き位置情報の割合が全体の 10% 以下の場合に, 教師有り学習手法より平均精度で 2% から 3% の性能が向上し, 精度の標準偏差が大きいことが明らかとなった. また, 提案手法で精度の標準偏差が大きいことは, ラベルの付け方次第で推定精度に大きな差がでるということであり, 実験の結果からラベルの付け方の指針を示した.

目次

図目次

表目次

第 1 章

序論

1.1 研究の背景

スマートフォンに代表される GPS 機能を搭載した機器の普及に従い, 最近では大量の位置情報を容易かつ安価に取得可能となっている. 位置情報とは対象がいつ・どこにいるのかという情報であり, 多くの場合は緯度・経度・時間の情報である. 得られた位置情報を用いて提供されるサービスを Location Based Service (LBS) と言う. その市場規模は 2011 年から 2017 年にかけて北米では 7 億 6000 万ドルだったものが 12 億 9500 百万ドルに, 欧州では 2 億 5000 万ユーロだったものが 8 億 2000 万ユーロにそれぞれ成長すると予想されている [日野 14]. 日本でも同様の増加傾向が予想されており, ビジネス面で今後成長が期待されている領域である.

位置情報は, ユーザーが興味のありそうな場所を推薦するシステム [ZZXY10, YYLL11, BZM12], 移動経路のプランニング [LWY+10, YZXW10], 人や物の移動の解析 [CCLS11, KIIF10], 移動パターンの発見 [PSR+13] など, 様々な研究やアプリケーションで用いられている. 位置情報に関する様々な研究やアプリケーションがある中で, 人の交通移動手段を利用する研究やアプリケーションが存在する. 交通移動手段とは車・自転車・徒歩などの移動形態のことである. 例えばアプリケーションの 1 つとして, 個人が環境に与える影響を計測するシステム [FDK+09, MRS+09] がある. このシステムでは, 個人の移動が環境へ与える影響を, 車や徒歩といった個人が使用した交通移動手段をもとに計算し評価する. そしてその情報をユーザーに提示し, 例えば自家用車の利用が多い人に対し, 公共交通機関の利用を促すことで, クリーンな交通移動手段の利用を促進している. 別の例として, Geolife[geo] というアプリケーションがある. このアプリケーションでは, 旅行の移動経路に関する位置情報をウェブにアップロードをすると, 実際に用いた交通移動手段を推定して, どのような交通移動手段で旅行を行ったのかを理解できるようになる. これによって思い出をより鮮明に思い出すことが可能となるだけでなく, 他の人が同じ場所を旅行をする際に, どの交通移動手段を選択すれば良いかの助けとなる. また観光統計データの 1 項目として交通移動手段が知りたい場合もある. 例えば, 日本では観光庁がアンケート方式で調査している統計データの 1 項目として交通移動手段の項目がある[*1]. この情報を利用することで, 外国人向けの案内をどの交通機関に重点的にすべきかなどの政策立案の助けとなる. 更には研究において, 例えば車の移動解析を行う際には, 車に関する位置情報のみを必要とする. そのため, 前処理として不必要な交通移動手段

[*1] http://www.mlit.go.jp/kankocho/siryou/toukei/shouhidoukou.html

の位置情報がデータに含まれている場合は除く処理が必要となる場合がある.

このように交通移動手段を知ることはアプリケーションとしても前処理としても重要な仕事である. 交通移動手段を知るには, ユーザーに直接通知してもらう他には, 加速度計やマイク, 気圧計などのセンサー情報を用いた推定 [ZP13, CCH+08] や, GPS などの位置情報 [ZCL+10] を用いた推定などの手段がある. 特に位置情報から交通移動手段を推定する研究分野では, 車や自転車などの交通移動手段を交通移動モードと言い, 交通移動モード推定とは, 位置情報から得られる移動軌跡データ上の任意の位置における交通移動モードを判定することである.

既存の交通移動モード推定に関する研究では, 推定方法として教師あり学習手法を使用した推定が行われている [ZCL+10, RMB+10, WHO+13]. そのため, 推定には学習に用いるラベル付き位置情報データが必要となる. ラベル付きデータの作成は, 移動軌跡を Web 画面上で確認しラベルを付与したり, 移動中に移動モードが変わった事を端末に逐次知らせるなどの手段で行われる. このように人手でラベル付きデータの作成が行われることに加え, データの性格上, 移動した本人しかラベル付の作業を行うことが出来ず, 第三者にラベル付け作業を任せづらい問題も存在するので, ラベル付きデータの作成には大きな時間的・金銭的コストが掛かる. 一方で, ラベルなし位置情報データの量は膨大である. よって, なるべく少ない量のラベル付きデータのみで十分な推定精度を達成することは, ラベル付け作業のコスト削減に対し有効となり重要である.

1.2 研究の目的

本研究では, ラベル付きデータの量が少ない場合でも, ラベル付きデータが十分にある場合と同等の推定精度で交通移動モード推定を行うことを目的とする. その目的を達成するために, 既存研究では教師あり学習手法を用いて推定を行っていた所を, 本研究では半教師あり学習手法を用いて推定を行う. 本研究の具体的な目的は, 第 1 に既存の半教師あり学習手法を交通移動モード推定に適用し, どの手法が交通移動モード推定に有効であるかを確認し, 第 2 に交通移動モード推定に有効である手法を改良することで, 推定の精度をより向上させると共に, どのデータに対しラベルを与えることが重要であるかを明らかとする.

1.3 本論文の構成

本論文の構成は次の通りである. 第 2 章では, 交通移動モード推定に関する先行研究をまとめ, 交通移動モード推定が一般的にはどのように行われていて, 先行研究ではどのような問題点を解決してきたのかを理解する. 第 3 章では, 推定のための半教師あり学習に関する先行研究を紹介する. 第 4 章では, 第 3 章で紹介した既存の半教師あり学習手法のうち 4 つ (Self-training model, Generative model, Co-training, Label propagation) を交通移動モード推定に適用し実験を行い, どの手法が有効であるかを確認する. 第 5 章では, 第 4 章での結果を踏まえ Label propagation を拡張した推定手法を提案し, その手法を用いた交通移動モード推定システムを構築する. 第 6 章では, 第 5 章で構築したシステムに関する評価実験を行い, 提案手法の有効性を確認すると共に, 依然として解決されていない問題点に関しても指摘する. 第 7 章では, 本研究全体のまとめを行い, 今後解決すべき問題点を述べる.

第 2 章

交通移動モード推定

本章では始めに, 本稿で使用する用語と交通移動モード推定の問題を定義する. 次に, 交通移動モード推定に関する関連研究を紹介し, 一般的な推定手法の流れを把握する.

2.1 問題定義

点 p とは (Object_ID, 緯度, 経度, 時刻) の 4 つの要素を持つ時空間内の点である (図 2.1). GPS による位置情報は, 緯度・経度・時刻・ID の情報を含むため, 位置 p は直接取得可能である. その精度は [gps] によると, 95% の精度で誤差 7.5 m 以内である.

軌跡 T とは同じ Object_ID を持つ点 p を時刻の昇順に並べたシーケンス $\langle p_1, p_2, ..., p_n \rangle$ であり, p_iと p_{i+1} の間は線形補間されている (図 2.1). また本稿では図 5.1 の T_i のように軌跡の始点を出発地, 終点を目的地とする. つまり人が建物に滞在している場合などに生じる点の集まりは軌跡とは定義しない.

セグメント S とは軌跡 T の部分シーケンス $\langle p_i, p_{i+1}, ..., p_{i+m} \rangle, 1 \leq i < i+m \leq n$ である (図 2.1).

交通移動モード (又はモード) とは車, 電車, バス, 徒歩といった人の移動形態 (図 2.2) のことであり, アプリケーションごとに事前に決定される. なお停止状態も交通移動モードの一種と見なすことがある. また交通移動モードが変化する点をチェンジポイントと定義する.

交通移動モード推定とは, 交通移動モードの集合 M と軌跡 T が与えられたとき軌跡上の任意の点での交通移動モードを推定することである.

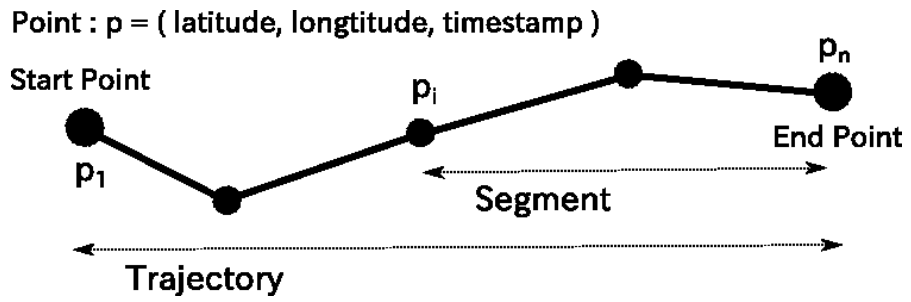

図 2.1: 点・軌跡・セグメント.

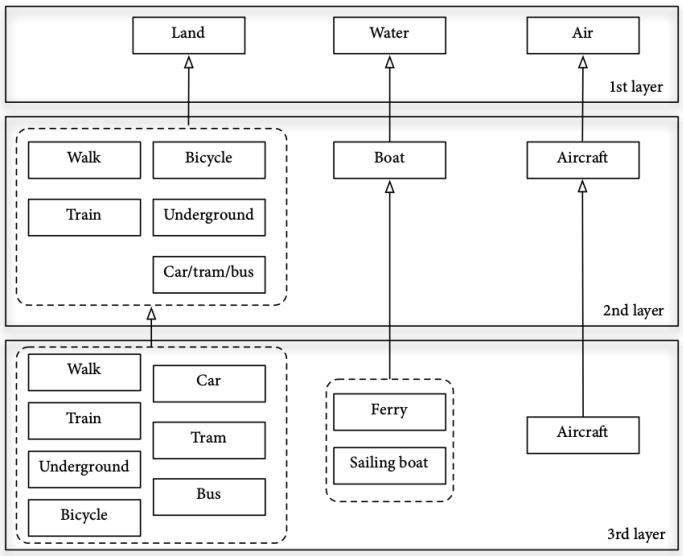

図 2.2: 交通移動モードの階層的表現.

[BLvO13]

2.2 関連研究

一般的な交通移動モード推定の流れは, 1) データ収集, 2) 前処理, 3) セグメンテーション, 4) 特徴量選択, 5) 推定の 5 つのフェーズに分けることができる. データ収集では, 位置情報の収集を行う. 前処理では生データから軌跡の抽出を行う. 具体的には, クラスタリングを行い, 点の集まりを発見し, クラスタ内の時刻が最も早い点と遅い点を軌跡の終点と始点にする. または電源が切れている場合や電波が通じない場合を想定し, 点と点の時間間隔がある一定時間以上であれば別の軌跡と見なす. 前処理としては他にデータのノイズを除去する為の Map Matching や Filtering の処理がある. なお軌跡抽出以外の前処理に関しては交通移動モード推定のスコープを若干外れる為, 詳しい説明はしないが [ZZ11] などに詳しい説明が載っている. 一般的な軌跡は, その中に複数の交通移動モードを含んでいる, セグメンテーションでは軌跡を単一のモードのみを含むセグメントへ切り分ける作業を行う. 特徴量選択では推定に用いるセグメントの特徴量を決定する. 推定では特徴量選択で選択した特徴量を基に, 一般的には教師あり学習手法の分類手法を用いて各セグメントがどのモードに属するのかを決定する. 以下では特に 1), 3), 4), 5) に関する関連研究を紹介する.

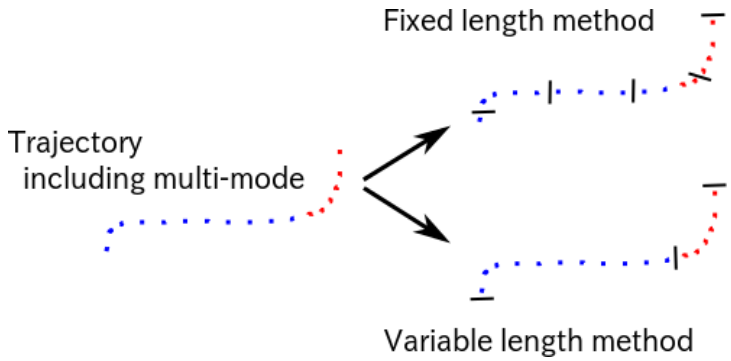

図 2.3: 2 種類のセグメンテーション方式.

2.3 データ収集

データ収集では位置情報の収集を行う．交通移動モードのラベル付き位置情報の収集方法はスマートフォンや専用の GPS 端末などの機器を持ち移動し, 移動中に交通移動モードが変化するごとに機器に通知したり, 移動後に Web 画面上で移動者本人がラベルを付与する方法が採用されている．ラベルなし位置情報の収集はスマートフォンや専用の GPS 端末などの機器を持ち移動し収集するだけでよい．

先行研究でのデータサイズは,65 人 10 ヶ月 [ZCL+10], 81 人で 2 週間 [BCTH12], 3 人で 1 ヶ月 [SVL+06], 6 人で 20 時間 [RBE+08] など論文によって様々である．

2.3.1 セグメンテーション

軌跡 T には複数の移動モードが含まれている場合が多い．よって軌跡 T を 1 単位としてモード推定を行うと, 複数のモードが単一のモードとして推定される．これは精度の面で望ましくない．そのためセグメンテーショでは軌跡 T をできるだけ単一の移動モードのみを含むセグメントに分割する．

関連研究におけるセグメンテーションの方法は大きく分けて次の 2 つの方法がある (図 2.3).

1. スライディングウィンドウを用いた固定長方式 [SVL+06, RBE+08, RMB+10, WCM10, BCTH12]
2. チェンジポイントを用いた可変長方式 [ZCL+10, STSA12]

本節では, この 2 つのセグメンテーションに関して詳しく説明する．

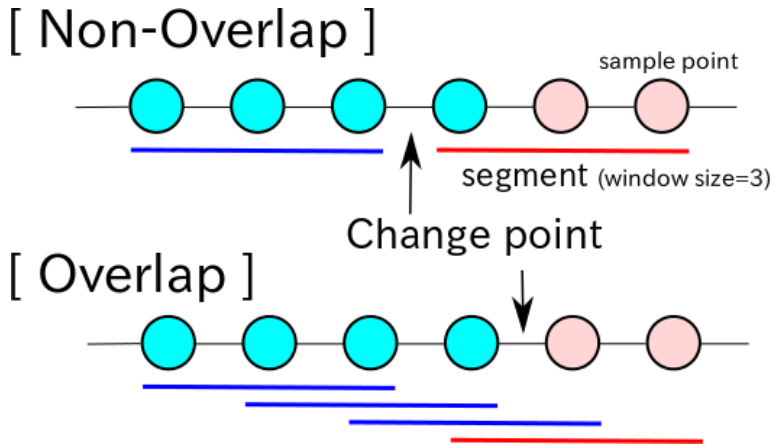

図 2.4: スライディングウィンドウ方式 (Overlap and Non-Overlap)

スライディングウィンドウを用いた固定長方式

固定長方式のセグメントは, 時間幅または点の個数を一定に固定したセグメントである (図 2.3 右上). この方式で軌跡を分割した場合, 短い固定長のセグメントが得られる. 同じモードの連続するセグメントはモード推定の後にマージされ最終的に単一のモードのみを含む長いセグメントが得られる. この方式ではチェンジポイントは推定の後に決定される.

関連研究では, ウィンドウが前のウィンドウとオーバーラップしない方式 [SVL+06, WCM10, RMB+10], オーバーラップする方式 [RBE+08, BCTH12] が存在する (図 2.4). またウィンドウサイズはサンプルサイズ固定の場合は 3 サンプル程度, 時間固定の場合は 1 秒から 3 秒 (サンプル数に直すと同様に 3 サンプル程度) が一般的である [SVL+06, WCM10, RMB+10, BCTH12].

オーバーラップする/しない場合でチェンジポイントの判定方法は異なる. ウィンドウがオーバーラップしない場合 (図 2.4 上) は, モードが変化したセグメントとセグメントの間をチェンジポイントとする. 例えば図 2.4 上では左のセグメントは青モード, 右のセグメントは赤モードと分類されているので, その間をチェンジポイントとする. 一方でオーバーラップする場合 (図 2.4 下) は, モードが変化したセグメント内の点と点の間をチェンジポイントとする. 例えば図 2.4 の下の様に, 4 つのオーバーラップセグメントが左から順に青・青・青・赤モードと推定された場合を考える. この場合にはモードが青と判定されたセグメントは 3 点中 2 点以上が青モードであり, モードが赤であると判定されたセグメントは 3 点中 2 点以上が赤モードであると考えられる. よってウィンドウサイズ 3 の場合は赤モードに変わったセグメントの左 2

表 2.1: 移動モードの遷移表.[ZCL+10]

Transportation Modes	Walk	Driving	Bus	Bike
Walk	/	41.0%	49.0%	9.0%
Driving	99.7%	/	0%	0.3%
Bus	98.7%	0.8%	/	0.5%
Bike	99.8%	0%	0.2%	/

点の間をチェンジポイントとする.

チェンジポイントを用いた可変長方式

可変長方式のセグメントは, Walk セグメントの始点と終点をチェンジポイントとして分割したセグメントである (図 2.3 右上). この方式では同じモードのセグメントはなるべく分割せずに, 長さを保つことを目的としているので, 理想的には推定の後にセグメントをマージする必要はない.

チェンジポイントの発見方法は 2 つの経験則に基づいている.

1. 乗り物から乗り物へのモード変化の間には必ず徒歩モードを挟む. よって徒歩モードのセグメントの始点と終点はチェンジポイントの可能性が高い.
2. 移動モードが変化する際には一度速度がゼロに近づき止まってから変化する.

なお Zheng ら [ZCL+10] は上で述べた経験則を実験的に示すために, 集めた軌跡データを元に図 2.1 の移動モードの遷移表を作成した. この表はあるモードからあるモードへの変化の割合を表している. 例えば Walk モードから Driving モードへの変化の割合は 41.0% である. この表によると徒歩から乗り物へモードが変化する確率と乗り物から徒歩へモードが変化する確率は高いが. 乗り物から乗り物へモードが変化する確率はいずれも 1% を切っている. これは, 1) の仮定がある程度の妥当性を持っている証拠となる.

この考え方に基づき可変長方式 [ZCL+10, STSA12] では, まず速度・加速度を基に Walk セグメントを推定する. そして Walk セグメントの両端をチェンジポイントとし, その点で軌跡をセグメントへと分割している. なおチェンジポイント発見の精度は [ZCL+10] では, Precision/Recall が 4 割/7 割程度とそれほど高くはない.

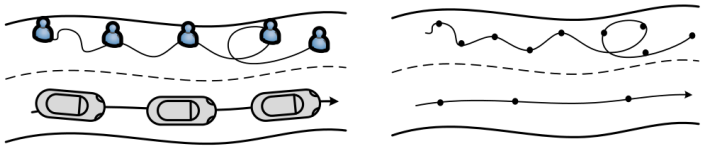

図 2.5: Heading Change Rate (HCR).

[ZCL+10]

2.3.2 特徴量選択

本節では推定で用いる特徴量について紹介する. 推定はセグメント単位で行われるので, 特徴量もセグメントの中から得られる量となる.

関連研究では, セグメント内の速度や加速度の平均・分散・最大値, セグメント長といった量が一般的には用いられている. 以下では関連研究の中で用いられている, それら以外の特徴量を紹介する.

Heading Change Rate (HCR) [ZCL+10, STSA12, SWYX11, GBGC12]

図 2.5 の様に, 典型的には自転車や車は人間ほど進行方向を変化させない. そこでセグメント内で進行方向の変化率がある閾値を超えた点の総数を $|P_c|$, セグメント長を $Distance$ として, Heading Change Rate(HCR) を次のように定義して特徴量として扱う.

$$HCR = |P_c| / Distance$$

Stop Rate (SR) [ZCL+10]

Zheng ら [ZCL+10] は, 実データをもとに速度の変化に関するグラフ (図 2.6) を作成した. このグラフによると各モードによってセグメント中での停止回数は異なる. そこでセグメント内で速度の値がある閾値以下になった点の総数を $|PS|$, セグメント長を $Distance$ として, Stop Rate(SR) を次のように定義して特徴量として扱う.

$$SR = |PS| / Distance$$

Velocity Change Rate (VCR). [ZCL+10]

図 2.6 から, 速度が大きく変化する回数も各モードによって異なる. そこでセグメント内で速度の変化率 $VRate = |V_2 - V_1| / V_1$ がある閾値以上になった点の総数を

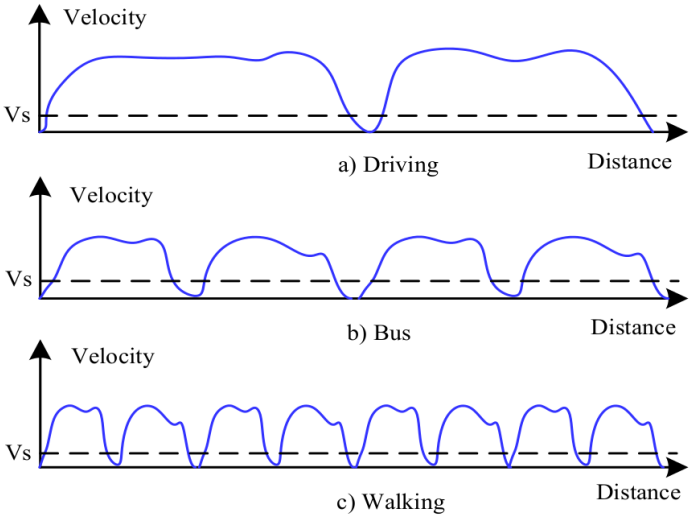

図 2.6: Stop Rate (SR).

[ZCL+10]

表 2.2: 推定アルゴリズム.

Naive Bayes [RBE+08, SWYX11]	Support Vector Machine [RBE+08, BCTH12]
Random Forest [SWYX11]	Decision Tree [MEBH08, RBE+08, ZCL+10, RMB+10]
Neural Network [GWB+09]	k-Nearest Neighbor [RBE+08]
Bayesian Net [SWYX11]	Conditional Random Field [ZCL+10]
Expert System [BLvO13]	Hidden Markov Model [AM06, RBE+08, RMB+10, WNB12]

$|P_v|$, セグメント長を *Distance* として Velocity Change Rate(VCR) を次のように定義して特徴量として扱う.

$$VCR = |P_v| / Distance$$

加速度センサーに関する特徴量. [WCM10, RBE+08, WNB12]

加速度センサーを用いた手法では, 加速度センサーから得られる加速度の平均・分散・最大値の他に, センサーから得た値を高速フーリエ変換 (FFT) して得られたパワースペクトル・エネルギーなどを特徴量として扱っている.

2.3.3 推定

関連研究における基本的な推定手法は教師あり学習手法を用いた分類である (表 2.2). 特徴数は GPS などの位置情報のみの場合は多くて 10 個程度. 加速度セン

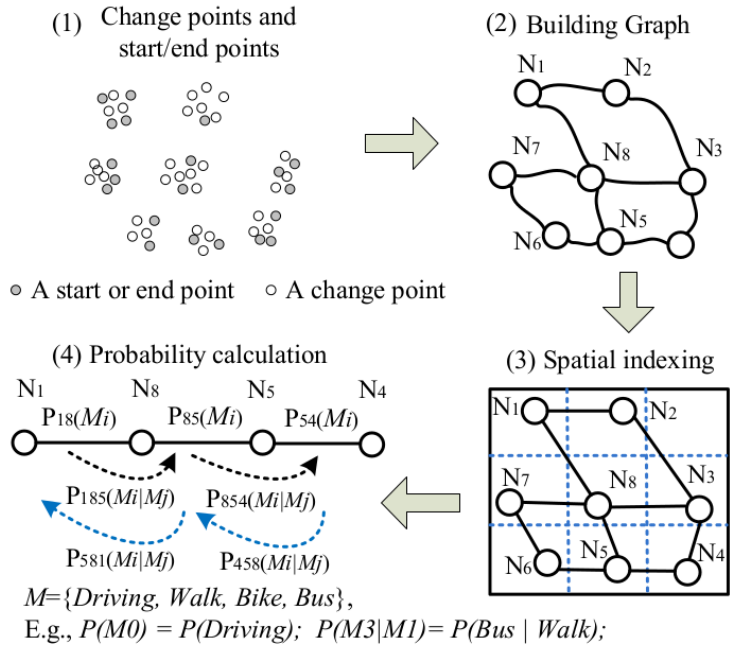

図 2.7: チェンジポイントを利用した後処理.

[ZCL+10]

サーを用いると 20 個から 70 個くらいと増加する.

以下では推定に関して, 1) 推定精度の向上, 2) モード数の増加という 2 つの視点から関連研究を紹介する.

推定精度の向上 [ZP13, RMB+10, ZCL+10]

表 2.1 で示されているように, 移動モードの遷移は確率が一定ではない. 徒歩から乗り物への遷移は起こりやすいが, 乗り物から乗り物への遷移が発生することは稀である. またある場所からある場所への空間遷移も移動モードごとに確率は異なる. 例えばバス停からバス停への遷移はバスモードである可能性が高い. 各セグメントを個々に分類するだけではこの遷移に関する情報は考慮されることはない. そこで遷移情報を用いて推定精度を向上している論文を以下で紹介する.

Zheng ら [ZCL+10] の手法では地図情報を用いる事なしにチェンジポイントをクラスタリングすることで, 場所から場所への遷移確率を考慮に入れた推定を行っている. この手法でのセグメンテーション方法はチェンジポイントを用いた可変長方式である. 特徴量集合を X, モードの集合を M とする. 1 段階目では決定木を用いてセグメントが各モード M_i へ属する確率 $P(M_i|X)$ を得る. 2 段階目では更に場所から場所への遷移を考慮した確率 $P(M_i|X, E_{ij})$ を計算する. まず訓練データ中の軌跡の始

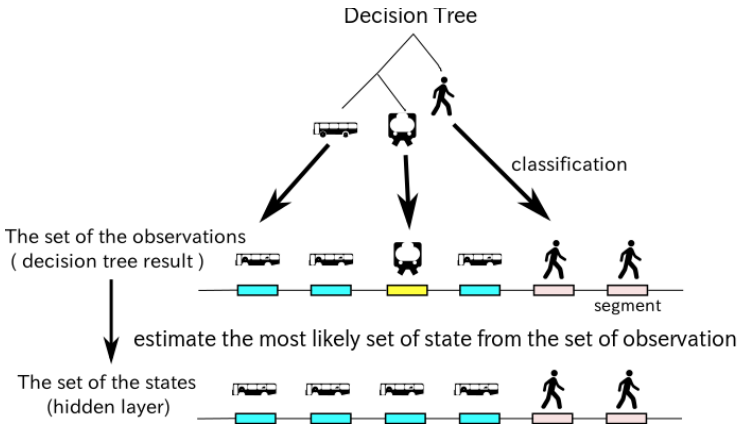

図 2.8: 隠れマルコフモデルによる後処理.

点・終点とチェンジポイントをクラスタリングする (図 2.7 左上). チェンジポイントをクラスタリングする為, この操作によって得られるクラスタ (ノード N_i) は現実世界ではバス停や駅といったモードを切り替える場所である可能性が高い. この操作によって直接地図情報を用いることなしに, バス停や駅に関する知識が得られる. 次にクラスタ間を結びグラフを作成する (図 2.7 右上). また計算の効率化の為に Spacial indexing を行う (図 2.7 右下). ノード $N_i - N_j$ 間のエッジ E_{ij} には, モード m_i ごとのノード間の遷移確率 $P(M_i|E_{ij})$ を予め訓練データを用いて計算する (図 2.7 左下). そしてベイズの定理を用いることで確率 $P(M_i|X, E_{ij}) = P(M_i|X)P(M_i|E_{ij})/P(M_i)$ が計算できる.

次に Zhang ら [ZP13] と Reddy ら [RMB+10] の論文を紹介する. この論文では隠れマルコフモデルを用いることでモード遷移を考慮した推定を行っている. この手法でのセグメンテーション方法はスライディングウィンドウを用いた固定長方式である. 固定長方式では可変長方式に比べセグメントが短い. よって渋滞や信号での停止のように一時的な状態の変化に対して間違えた分類結果を出力する可能性がある. 例えば図 2.8 では実際はバスと分類されなければならないセグメントが電車と分類されている. この問題を解決する手段の 1 つとしてこれらの論文では隠れマルコフモデルを用いている. 隠れマルコフモデルの問題とは実際に観測された観測系列から, その背後に存在する真の状態系列を推定する問題である. この手法では 1 段階目は決定木を用いてセグメントを分類する. 2 段階目ではその分類結果を隠れマルコフモデルの観測系列として, その観測系列を最も良く生成する真の状態系列を推定し, モード推定の結果とする.

モード数の増加. [SWYX11, BLvO13, GCBL11]

モード数が少なく徒歩・自転車・車といったモードを分類したい場合は, 速度や加速度の特徴量によって比較的精度よく分類が可能である. しかしモード数が増加すると, 車とバスのように速度と加速度といった特徴量だけでは誤分類が起きる可能性がでてくる. そこでモード数を増加させる場合は GPS データ以外の情報も用いることが必要となる. 以下では GPS データ以外の情報も用いてモード推定をしている論文を紹介する.

Geographic Information System (GIS) とは様々な地理情報を参照・検索できるシステムである. この GIS から得られる量を特徴量として扱う関連研究がある [SWYX11, BLvO13, GCBL11]. これらの論文では, 最も近い線路/バス停との距離や地下鉄の出口からの距離といった量を特徴量として扱った. また走行中のバスの位置との距離といった動的に得られる情報による特徴量も扱われている. GIS を用いることで車とバスのように似たような特徴を示すモードの推定がより高精度で行える可能性がある.

Biljecji らは [BLvO13] で図 2.2 のようにモードを階層的に表現し多段階で推定している. 例えば図 2.2 をみるとレイヤー 2 では car/tram/bus を 1 つのモードとして扱っていることがわかる. 階層化の利点は階層的にモードを定めておくことで, 用途に合わせたモードの粒度で推定が可能となる点である.

2.3.4 関連研究のまとめ

表 2.3 は関連研究を整理した表である. この表の列は左から、論文の発表年、参考論文、用いたセンサーデバイス、推定時に使用したアルゴリズムの中で最も精度の高いアルゴリズム、GIS の使用の有無、推定精度、位置情報のサンプル率、モード数、特徴量数となっている。なお表の Accuracy に関しては, データサイズやモード数といった評価の土台が全ての手法で異なるため単純な比較は出来ないことに注意したい.

関連研究における交通移動モード推定手法はほぼ全てが教師有り学習による分類であり, 半教師あり学習や教師なし学習による推定を行った研究は確認されなかった. GPS による位置情報のみを使用した手法の精度は 80%〜90% 程度, GPS による位置情報の他に GIS や加速度計を使用した場合は 90%〜94% 程度である. 分類アルゴリズムによる推定精度の差はそれほど見られないが, 決定木やランダムフォレストが多く使用されている. また分類するモードは車, 徒歩, 自転車, 電車の 4 種類をベー

スとして, モードが多いものは車を更にタクシーとバスに分割したり, 静止 (stay) モードを追加するなどしている.

2.3.5 関連研究の問題点

交通移動モード推定ではラベル付け作業は移動した本人しかすることができず, 第3者に任せることができないため, ラベル付き位置情報を収集しづらいという問題がある. しかし関連研究では, 性能評価の際に訓練データの量として 7 割 [ZCL+10], , 9 割 [RMB+10, BLvO13, WHO+13] など半分以上のデータを用いて訓練を行っている. 実際には, ラベル付きデータに対しラベルなしデータは非常に大量にあることが想定されるため, ラベルが少ない場合にどの程度の推定精度がだせるかに関する議論は必要である. また, どの位置情報に対してラベルを付けることが, より早い推定精度の向上に重要であるかを考える必要もある.

表 2.3: 関連研究一覧

Year	Paper	Sensor	Algorithm	GIS	Accuracy	Sample Rate	# Modes	# Features
2006	[SVL+06]	GSM[*1]	Boosting	×	85%	—	3	7
	[AM06]	GSM	HMM[*3]	×	80%	—	3	—
2008	[MEBH08]	GSM,Wifi	DT[*4]	×	88%	1sample/2s	3	4
	[RBE+08]	GPS,Acc[*2]	DT, HMM	×	93.2%	1sample/1s	5	54
2009	[GWB+09]	GPS	NN[*5]	×	91.2%	—	3	8
2010	[ZCL+10]	GPS	DT	×	78%	1sample/2s	4	9
	[RMB+10]	GPS,Acc	DT, HMM	×	93.6%	1sample/1s	5	5
	[WCM10]	Acc	DT	×	70.7%	—	6	23
2011	[SWYX11]	GPS	RF[*6]	○	93.5%	1sample/15s	6	8
	[GCBL11]	GPS	RuleBase	○	82.6%	—	5	—
	[XSG+11]	—	HMM	×	90.5%	—	3	1
2012	[WNB12]	GPS,Acc	HMM	×	76%	—	8	77
	[GBGC12]	GPS	NBT[*7]	×	73%	—	3	3
	[BCTH12]	GPS	SVM[*8]	×	88%	1sample/60s	6	4
2013	[BLvO13]	GPS	ES[*9]	○	91.6%	6.5sample/1s	10	9
	[ZP13]	Acc	DT, HMM	×	88%	—	5	11
	[WHO+13]	GPS	RF[*6]	○	84%	—	5	8

[a] Global System for Mobile Communications
[b] Accelerometer
[c] Hidden Markov Model
[d] Dicision Tree
[e] Neural Network
[f] Random Forest
[g] Naive Bayes Tree
[h] Support Vector Machine
[i] Expert System

第 3 章

学習と予測

本章では, 推定の手段として一般的に使用される教師あり学習・教師なし学習についてまずは述べる. その後, 少量のラベルのみで効率的に推定を行う手法として, 半教師あり学習による推定手法を紹介する.

3.1 機械学習一般

一般的な機械学習は, 教師あり学習と教師なし学習の 2 つに分類できる. 本節ではそれら 2 つに関して簡単な説明を行う. なお詳細に関しては [HTFF05, Bis06] などを参考にされたい.

3.1.1 教師あり学習

教師あり学習 (Supervised Learning) の目的は, 入力に対応する N 個の d 次元特徴量ベクトル $\mathbf{x}_i (i = 0, ..., N-1)$ と, 各入力に対応する出力の値 $y_i (i = 0, ..., N-1)$ が与えられた時, 未知の入力に対する出力の値を予測することである. 予測のための規則を生成する際に使用するラベル付きデータ集合 $\{(\mathbf{x}_0, y_0), ..., (\mathbf{x}_{N-1}, y_{N-1})\}$ のことを訓練データ (Training Data) と言い, 学習とは入力 $\mathbf{x}$ に対して, 出力 y を決定する規則 $f : \mathbf{x} \mapsto y$ を決定することである.

教師あり学習のタスクは, 出力の形式によって大きく 2 つに分類できる.

1. 分類 (Classification)
   出力 y_i は離散クラス集合 (ラベル集合)$\mathcal{Y} = \{l_1, ..., l_c\}$ のいずれか一つの値をとる.
2. 回帰 (Regression)
   出力 y_i は連続値をとる.

評価に関しては, 教師あり学習では予測が成功か失敗の明確な尺度が存在するので, それを利用し評価を行う.

3.1.2 教師なし学習

教師なし学習 (Unsupervised Learning) では, 教師あり学習と異なり入力に対応する N 個の d 次元特徴量ベクトル $x_i (i = 0, ..., N-1)$ のみが与えられる. そのため, 個々のデータをどのように扱えば良いかといった指示はない. 教師なし学習では, 一般的に次の 3 つのタスクのいずれかを目的としたものが多い.

1. クラスタリング (Clustering)
   N 個の入力を, グループ (クラスタ) に分ける.
2. 外れ値検知 (Outlier Detection)
   多数派とは異なる, 少数のデータ点を発見する.
3. 次元削減 (Dimensionality Reduction)
   入力データの特徴を保持しつつ, データの次元をより低くする.

教師なし学習では, 出力結果が妥当か否かの明確な尺度が存在しないことが多い. そのため, 評価にはヒューリスティックな議論が必要となることもある.

3.2 半教師あり学習

半教師あり学習 (Semi-Supervised Learning) とは, 教師あり学習と教師なし学習の中間にあたる学習方法である. 半教師あり学習ではラベルありデータ $\{(x_1, y_1), ..., (x_l, y_l)\}$ とラベルなしデータ $\{x_{l+1}, ..., x_n\}$ の両方を使用して推定を行う. なお, 一般的にはラベルありデータはラベルなしデータに対し非常に少ないものとする ($l \ll n$). よって半教師あり学習には, 教師あり学習にラベルなしデータを加えるという方向性と, 教師なし学習にラベルありデータを加えるという 2 つの方向性がある.

まず, 半教師あり学習の利点について述べる. 教師あり学習では, 全てのラベルありデータを用いて訓練を行い, 未知の入力に対する推定を行った. しかし, ラベルありデータの量が少ない場合は, 十分な推定精度が得られないといった問題が生じる. その場合, ラベルありデータの数を増やせば問題は解決されるが, ラベル付け作業は基本的に人手で行われるため, 大きな金銭的・時間的コストが掛かる場合が多い. またデータによっては, ラベル付の際に専門知識が必要となる場合もあり, そのような場合にはラベル付のコストが更に増大する. 一方で, ラベルなしデータは大量かつ無料で得られる場合が多い. そのため, 少量のラベルありデータと大量のラベルなしデータを組み合わせることで, ラベル付きデータが少ない場合に精度の向上が図れれば, コストの問題にも対応可能となる.

このような利点は存在するが, ここでラベルなしデータから規則 $f : \mathbf{x} \mapsto y$ に関する何らかの学習が可能であるのかという疑問が生じる. 明らかに, ラベルなしデータは規則 $f : \mathbf{x} \mapsto y$ に関して何の情報も持たないので, ラベルなしデータを共に用いたとしても, ラベルありデータから得られる情報のみを使用した学習しか出来ない. ラベルなしデータを有用に用いるために, 半教師あり学習ではラベルありデータとラベ

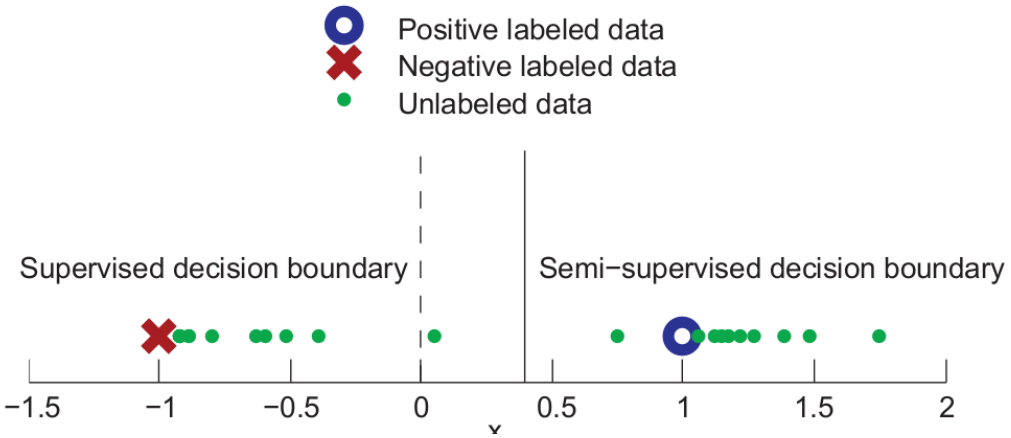

図 3.1: 半教師あり学習と教師あり学習手法の違いを示す簡単な例. [ZG09]

ルなしデータを結びつける仮定が必要となる. 例えば, 図 3.1 では, 2 つのラベルありデータ (青丸と赤十字) のみでを用いて教師あり学習で推定した場合, この 2 点がクラスを代表する点となり決定境界は 2 点の中央の位置にある点線になる. 一方で, ラベルなしデータ (緑丸) も含めた場合を考えると, データは 2 つのグループに分かれていることが観察できる. そこで, 各クラスが $p(\mathbf{x}|y_i)(i = 0, 1)$ がガウス分布に従うという仮定をおくと, 2 つのラベルありデータは最早クラスを最も良く表しているプロトタイプとはならなくなり, ラベルなしデータも用いた半教師あり学習で推定した場合は, 決定境界は実線の位置にずれる.

このように, 半教師あり学習ではラベルなしデータとラベルありデータを結びつけるための仮定が非常に重要となる. そのため, 仮定に誤りがある場合はラベルなしデータを共に使用したからといって, 必ずしも教師あり学習より良い精度がでるわけではないということに注意するべきである. 以下では, 既存の半教師あり学習アルゴリズムを, どのような仮定の下で使用されるべきかに注目し紹介する.

3.2.1 Self-Training Model

Self-training Model(以下では STM)[ZG09, MCJ06, RHS05] は, まずラベルありデータを使い規則 f を学習し, f を用いラベルなしデータを予測する. その後, 予測の信頼度が高いと判断されたデータをラベルありデータに加えて, f を再学習するというステップを繰り返すことで推定を行う方法である.

このモデルでは, 予測の信頼度が高い結果は正しいという仮定の下で半教師あり学習が行われる.

利点は, 既存の分類器のラッパーを書くだけで良くシンプルかつ実装が簡単という点であり, クラスが Well-separated なクラスターを形成している場合には推定は比較的上手くゆく. 一方で, 早い段階で間違ったラベルを付けてしまうと, 連鎖的に間違いが広がり, 推定精度が著しく低下する可能性があったり, 信頼度に対する閾値の設定方法が難しいなどの欠点がある.

3.2.2 Generative Model

Generative Model[RV95, ZG09] は, 背後に何らかの確率モデルが存在するという仮定をおくことで, 推定をラベルなしデータも使用して行う手法である.

ラベルありデータのみでパラメータの最尤推定を行う場合は, ラベル付きデータ $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)\}$ を用いて $\log(D|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(\mathbf{x}_i|y_i, \theta)$ の最大化を考えたが, Generative Model ではこれにラベルなしデータ $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l), \mathbf{x}_{l+1}, ..., \mathbf{x}_{l+u}\}$ も合わせ, $\log(D|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta)p(\mathbf{x}_i|y_i, \theta) + \sum_{i=l+1}^{l+u} \log p(\mathbf{x}_i|\theta)$ の最大化を考える.

利点は, ラベルなしデータの役割が明確であり, 確率モデルを使用し明確にモデル化出来る点である. 一方で, 欠点はモデルの仮定が正しくない場合はラベルなしデータによって, 寧ろ推定精度が悪化する可能性がある点である.

3.2.3 Co-Training

Co-training[BM98, ZG09] では, 特徴量を 2 つの視点により 2 分割 $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]$ し, それぞれの特徴量で学習器 $f^{(1)}, f^{(2)}$ を構築する. そして, それぞれの学習器が互いに推定結果を教えあうことで学習を行う. 例えば, Web ページの分類を行う場合を考える. Web ページの特徴量として画像による特徴量とテキストによる特徴量を使用するとした場合, 特徴量を画像による特徴量とテキストによる特徴量の 2 つに分割し, 画像の学習器とテキストの学習器を構築する. そして, それぞれが互いに信頼度が高い分類結果を教えあうことで, 徐々にラベルありデータを増やし推定を行うといった形である.

Co-training の仮定は 2 つ有り, 1) 分割後の特徴量は, 十分なラベルデータがある場合に, 各特徴量のみで十分な推定精度を出せる学習器が構築可能である, 2) 分割後の特徴量は, それぞれラベルが与えられたもとで条件付き独立である ($p(\mathbf{x}^{(1)}|y, \mathbf{x}^{(2)}) = p(\mathbf{x}^{(1)}|y), p(\mathbf{x}^{(2)}|y, \mathbf{x}^{(1)}) = p(\mathbf{x}^{(2)}|y)$).

利点は, self-training ほど間違いに敏感ではない点や, 既存の学習器に対してラッパーを書けば良いだけなので, 実装が容易である点などである. 一方で欠点は, 互い

に条件付き独立であるような分割が容易に見つからない場合もあるので, 使用場面が限定されることや, 両方の特徴量を使用して一つの学習器を構築した方が精度が良くなる可能性がある場合がある点などである.

3.2.4 Graph-Based Semi-Supervised Learning

Graph-based 半教師あり学習 [ZG09] では, 重みが大きな (類似度が高い) エッジで繋がっているノードは, 同じラベルであろうという仮定 (特徴量空間中で近い点は似ているという仮定) をおきラベルなしデータの推定を行う.

基本的な推定方法は, ラベルありデータ $\mathbf{X}_l$ とラベルなしデータ $\mathbf{X}_u$ をノードとするグラフを作成し, ノード間の重みを何らかの関数 (カーネル関数) で設定する. 例えば, 重みはばユークリッド距離を用いた重み $w_{ij} = \exp(-\alpha \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2)$ であったり, k 近傍を考え $w_{ij} = 1$(自身から近い k 番目以内の点), $w_{ij} = 0(otherwise)$ などがある. そして, ラベルありデータノードからグラフを伝搬させラベルなしノードのラベルを推定する.

Graph-based な方法の利点は, 同じクラスに属する点は, 特徴量空間中で近くに存在するという仮定が当てはまるデータに対しては上手くゆく点であり, 欠点はグラフが大規模になる可能性があるので空間・時間計算量が大規模になる場合には現実的な時間で終わらない可能性が出てくる点である.

3.2.5 Semi-Supervised Support Vector Machine

Semi-Supervised Support Vector Machine (S3VM) [Joa99, BD$^+$99, ZG09] は SVM を半教師あり学習に拡張したものである. S3VM では, 特徴量空間内の決定境界付近では, データの密度が低いだろうという仮定のもとで, ラベルなしデータの密度が高い場所では決定境界を引きにくくすることで, ラベルなしデータも利用した半教師あり学習をしている. そのため, 実際の決定境界が密度の高低とは関係ない場所に引かれる場合には性能が悪化する場合がある.

3.2.6 まとめ

この節では幾つかの代表的な半教師あり学習手法を簡単に紹介したが, 全ての手法に共通することは, ラベルなしデータを有効に利用するためにはデータに関して何らかの仮定をおく必要があるという点である. よって半教師あり学習は教師あり学習の拡張というより, 教師なし学習の拡張と捉えた方が良い場合が多い. 半教師あり学習

では, その仮定自体が各手法の特徴を表しているため, データを注意深く観察し, 正しい仮定のもとで使用する必要がある.

この節で挙げた半教師あり学習の手法は代表的なものを挙げたにすぎず, 他にも様々な手法があるので, 詳しくは [See00, ZG09] などを参考にされたい.

3.3 Label Propagation

Label Propagation (以下では LP)[ZG02, ZGL$^+$03] は Graph-based 半教師あり学習手法の一つである. LP では特徴量間の近さをエッジとしたグラフを作成し, 既知のラベル情報を伝搬させることで, 未知のラベルを推定する. 以下では LP の簡単な解説を行う.

3.3.1 問題設定

ラベルありデータを $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)\}$, クラスラベルを $\mathbf{Y}_L = \{y_1, ..., y_l\}$ とする. Y_L は観測可能であり既知である. C をクラス数とし, クラスラベル Y_L の中には全てのクラスが出現するものとする. ラベルなしデータを $\{(\mathbf{x}_{l+1}, y_{l+1}), ..., (\mathbf{x}_{l+u}, y_{l+u})\}$ とし, クラスラベルを $\mathbf{Y}_U = \{y_{l+1}, ..., y_{l+u}\}$ とする. ただし Y_U は観測出来ないので未知である. 一般的に $l \ll u$ となる. また $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_{l+u}\}(\mathbf{x}_i \in R^D)$ とする.

LP の目的は $\mathbf{X}$ と $\mathbf{Y}_L$ から, $\mathbf{Y}_L$ を推定することである. その LP では, 全てのデータを用いグラフを作成し, ラベルありデータからエッジを通じてラベルを伝搬させてゆき, ラベルなしデータのラベルを推定する.

近い点 (類似している点) 同士は同じラベルであろうという仮定に基づき, LP ではまず $\mathbf{X}$ の各点をノードとするグラフを作る. エッジは点 $\mathbf{x}_i$ と点 $\mathbf{x}_j$ の近さ (類似度) を表す重み w_{ij} を持っている. w_{ij} は問題に合わせて様々な形を取ることができるが, 例えばユークリッド距離に従った重み $w_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$(Radial Basis Function) や, k 近傍の点に注目した重み $w_{ij} = 1$(j が i の k 近傍点である), $w_{ij} = 0$(それ以外) などがよく使用される.

$(l+u) \times (l+u)$ 遷移確率行列 $\mathbf{T}$ は, $T_{ij} = P(j \to i) = \frac{w_{ij}}{\sum_{k=1}^{i+u} w_{ij}}$ であり, ノード j から i への遷移確率を表している. $(l+u) \times C$ のラベル行列 $\mathbf{Y}$ は, $Y_{ij} = P(C_j|\mathbf{x}_i)$ であり, $\mathbf{x}_i$ に対しラベル $C_j(j = 1, ..., C)$ がどの程度の確率であり得るかを表している. ラベルありデータはラベルが既知であるので, クロネッカーの δ を用いて $Y_{ic} = \delta(y_i, c)$ となる. ラベルなしデータに対しての初期値は, 最終的な結果には関係

ないことが後に示されるので, ランダムで良く特に重要ではない.

3.3.2 アルゴリズム

LP では次の 3 つのステップを繰り返すことで, ラベルを推定する.

1. $\mathbf{Y} \leftarrow \mathbf{TY}$ によってラベルを伝搬させる.
2. $\mathbf{Y}$ を行方向に正規化する.
3. $\mathbf{Y}$ のラベルありデータ部分を固定し, $\mathbf{Y}$ が収束するまで以上を繰り返す

ステップ 1 では, 1 ステップ分ラベルを伝搬させている. ステップ 2 では, $\mathbf{Y}$ が確率を表すように行方向に正規化している. ステップ 3 では, ステップ 1 でラベルデータの部分が $Y_{ic} = \delta(y_i, c)$ ではなくなっているが, ラベルデータの部分は $Y_{ic} = \delta(y_i, c)$ であるとわかっているので, $Y_{ic} = \delta(y_i, c)$ となるように置き換える. そして, $\mathbf{Y}$ が収束するまでこの作業を繰り返す.

ここで問題となるのは, $\mathbf{Y}$ の収束性である. それを証明するため, $\mathbf{T}$ を行方向に正規化した行列を $\bar{\mathbf{T}}$ とし, $\mathbf{Y}$ を上から l 行目までをラベルありデータとした $l \times C$ 行列 $\mathbf{Y}_L$ として, $l+1$ 行目から最後までをラベルなしデータとした $u \times C$ 行列 $\mathbf{Y}_U$ とする. さらに $\bar{\mathbf{T}}$ を次のようにする.

$$\bar{\mathbf{T}} = \begin{bmatrix} \bar{\mathbf{T}}_{ll} & \bar{\mathbf{T}}_{lu} \\ \bar{\mathbf{T}}_{ul} & \bar{\mathbf{T}}_{uu} \end{bmatrix} \tag{3.1}$$

$\mathbf{Y}$ の上側である $\mathbf{Y}_L$ は, ステップ 3 の操作があるため変化しない. よって, 興味のある部分は $\mathbf{Y}_U$ の部分となる.

$$\mathbf{Y}_U \leftarrow \bar{\mathbf{T}}_{uu}\mathbf{Y}_U + \bar{\mathbf{T}}_{ul}\mathbf{Y}_L \tag{3.2}$$

であり, ステップを繰り返すと,

$$\mathbf{Y}_U = \lim_{n\to\infty} \bar{\mathbf{T}}_{uu}^n \mathbf{Y}^0 + \left[\sum_{i=1}^{n} \bar{\mathbf{T}}_{uu}^{i-1}\right] \bar{\mathbf{T}}_{ul}\mathbf{Y}_L \tag{3.3}$$

となる. [ZG02] より, この式の前半部分は 0 に収束することがわかるので, 結局 $\mathbf{Y}_U$ は,

$$\mathbf{Y}_U = \left(\mathbf{I} - \bar{\mathbf{T}}_{uu}\right)^{-1} \bar{\mathbf{T}}_{ul}\mathbf{Y}_L \tag{3.4}$$

となる. つまり, 実際には $\mathbf{Y}_U$ を求める為に収束するまでステップを繰り返す必要はなく, 式 3.4 を解けば良い.

収束後の $\mathbf{Y}_U$ の i 行 j 列は, データ $\mathbf{x}_i$ がクラス j へ属する確率を表している. よって, クラス分類の場合は, 例えば最も確率が高いクラスをラベルとして与えるといった方法がある.

3.3.3 まとめ

Label propagation は特徴量空間内で近さをエッジとしたグラフを作成し, ラベルありデータの情報をグラフ全体に伝搬させることで, ラベルなしデータのラベルを推定する方法であった. この手法では, 同じクラスに属するデータは特徴量空間中で近くに存在するという仮定がおかれているため, そのような特徴をもつデータについては上手く動作する. また, データ間の近さ (重み) の定義が重要となる.

第 4 章

既存の半教師あり学習手法に関する実験

交通移動モード推定の先行研究において, 半教師あり学習手法を用いた研究は確認されなかった. そこで, この章では, 5 章で提案するシステムにおいて, どの半教師あり学習手法を使用するべきかの確認として, 前章で述べた既存の半教師あり学習手法のうち 4 つを, 実際に実装し交通移動モード推定を行う予備実験を行った.

実験では, 既存の半教師あり学習手法を用いた推定方法として, 1) Self-Training Model, 2) Generative Model, 3) Co-Training, 4) Label Propagation (Graph-based semi-supervised) の 4 種類を交通移動モード推定に適用し, 比較対象として教師あり学習手法の一つであるランダムフォレスト (以下では RF)[Bre01] と共に実験を行った.

実験環境や評価データ・評価方法に関しては 6 章を参考. 実験では, 特徴量は速度の平均・標準偏差, 停止点率, 距離/時間, 線路・道路とのマッチ率の 6 種類を採用し, セグメンテーション方法は時間分割の時間幅 60 s を採用している, また推定の部分以外の処理は次章のシステムと同じである (詳しくは 5 章を参照).

4.1 Self-Training Model に関する実験

推定に Self-Training Model(STM) を用いた実験を行う.

ラベルありデータの集合を $\mathcal{L}$, ラベルなしデータの集合を $\mathcal{U}$ とする. まず $\mathcal{L}$ を用いて, 規則 f を学習する. 学習アルゴリズムとして本実験では RF を採用する. 規則 f を用い, $\mathcal{U}$ の各データ u を推定する. 推定の信頼度が高いと考えられるデータ, 本実験では事後確率 $p(c|\mathbf{x})$ が閾値 α を超えたデータは正しいと仮定し, $(u, f(u))$ を $\mathcal{L}$ へ追加する. 以上を $\mathcal{U}$ が空になるまで繰り返す. (アルゴリズム 1)

Algorithm 1: Self-Training Model

Data: Labeled data $(\mathbf{x}_i, y_i)_{i=1}^{l}$, unlabeled data $\mathbf{x}_{j\,j=l+1}^{l+u}$

initialization $L = (\mathbf{x}_i, y_i)_{i=1}^{l}$, $U = \mathbf{x}_{j\,j=l+1}^{l+u}$;

while U **が空ではない**. **do**

ランダムフォレストを用い, L から f を学習する;

f を U に適用する;

U の中で事後確率が $p(c|\mathbf{x}) > \alpha$ であるデータを $\{\mathbf{x}, f(\mathbf{x})\}$ とし, L に加える;

end

4.1.1 実験結果

初期のラベル付き軌跡の数を 8 本 (全体のおよそ 1%), 28 本 (全体のおよそ 5%), 60 本 (全体のおよそ 10%), 236 本 (全体のおよそ 40%) と変化させ, $\alpha = 0.9$ として実験を行った結果が表 4.3 である. 結果が収束するまでのループ回数, つまり U が空になるまでのループ回数は, ラベル付き軌跡数が 8 本, 28 本, 60 本, 236 本の各場合に対し, 平均して 58.8 回, 73.1 回, 78.9 回, 91.3 回である.

表 4.1: Self-Training Model を用いた推定結果.

ラベルの割合およそ 1%	RF	STM	ラベルの割合およそ 5%	RF	STM
精度	71.39	70.54	精度	77.58	78.32
標準偏差	6.38	7.93	標準偏差	2.42	1.25
ラベルの割合およそ 10%	RF	STM	ラベルの割合およそ 40%	RF	STM
精度	79.56	80.32	精度	81.57	82.31
標準偏差	0.95	1.03	標準偏差	0.91	1.39

4.1.2 考察

実験の結果, self-training model を用いた推定ではラベルありデータの割合が全体の約 1% と, 特に割合が低い場合は精度が下がっているが, ラベルありデータの割合が 5% 以上の場合では, ラベルありデータの割合に関わらず, 僅かではあるが一定の精度上昇が観測できる. ラベルありデータが約 1% の場合に精度が悪化している原因の一つとしては, ラベルありデータが少ないことが原因で早い段階で推定を誤ると連鎖的に推定を誤り, 結果として著しく推定精度が下がってしまうことが考えられる. そのような場合に実験では, RF に比べて 10% 程度の推定精度の低下が観測された. また, アルゴリズムが収束するまでのループ回数が 60 回から 90 回程度であるため, 比較対象の RF に比べて計算時間は 60 倍から 90 倍程度掛かり, 計算時間の点では不利である.

4.2 Generative Model に関する実験

Generative Model を用いて交通移動手段推定を行うための, 調査と実験を行う.

Generative Model を用いる為には, 背後に存在する確率分布を適切に仮定する必要がある. そこで, 特徴量空間内にデータがどのように分布しているかに関する調査が, まずは必要となる. 図 4.1 は特徴量として, 速度の平均・標準偏差, 停止点率, 距離, 線路・道路とのマッチ率の 6 種類 (詳しくは 5.5 節を参照) を白色化 [Bis06] した結果を, 可視化した散布図行列である. なお, このプロットは 6 次元空間のデータを各特徴量ペアごとに 2 次元に射影したものなので, 6 次元空間中のデータの配置を正確には把握は出来ないことには注意したい.

可視化した結果を観察すると, モードが同じであれば, 特徴量空間中で近くに存在するということは言えそうであるが, 分散の度合いはモードごとに異なっている. また, 1 つのモードが, 1 つのクラスタを形成しているわけではなさそうである. 4 つのモードの中でも特に bike(水色の丸) に関しては, 停止点率 (sr) が bike を辛うじて分離できそうな特徴量であるように観察できるが, それでも良い精度がでるとは考えづらい. 更に, 各モードの分布がよく知られている確率分布 (例えば多次元ガウス分布) に従っているという仮定をおくことも難しいと考えられる.

可視化結果の観察より, 良い確率モデルを仮定することが難しいため, Generative model を用いた推定は適さないと考えられる. しかし, 参考の為に混合数がモード集合の要素数 (実験では要素数 4) と等しい混合ガウス分布を仮定し, ラベルなしデータとラベルありデータの両方を使用し混合ガウス分布のパラメータ推定を EM アルゴリズムで行う Semi-supervised Gaussian Mixture Model(SGMM)[RV95] を用いて交通移動モードを推定する実験を行った. なお混合ガウス分布と EM アルゴリズムについての詳細は C を参考にしてもらいたい.

4.2.1 実験結果

SGMM ではラベル付きデータの量が少ない場合には, 逆行列計算が出来ないために手法自体が利用不可能であった. またラベルありデータの割合をを全体の 50% まで増やした場合でも精度は 64% であった.

4.2.2 考察

実験の結果, 少なくとも背後に単純な混合ガウス分布が存在すると仮定することは不適切であることが示された. そのため, この手法を用いるには 1 つのモードを複数の確率分布で表し, 更にその混合分布の混合を考えるといった方向性をもった手法を考えるか, 混合ガウス分布以外の適切な確率モデルを発見する必要がある.

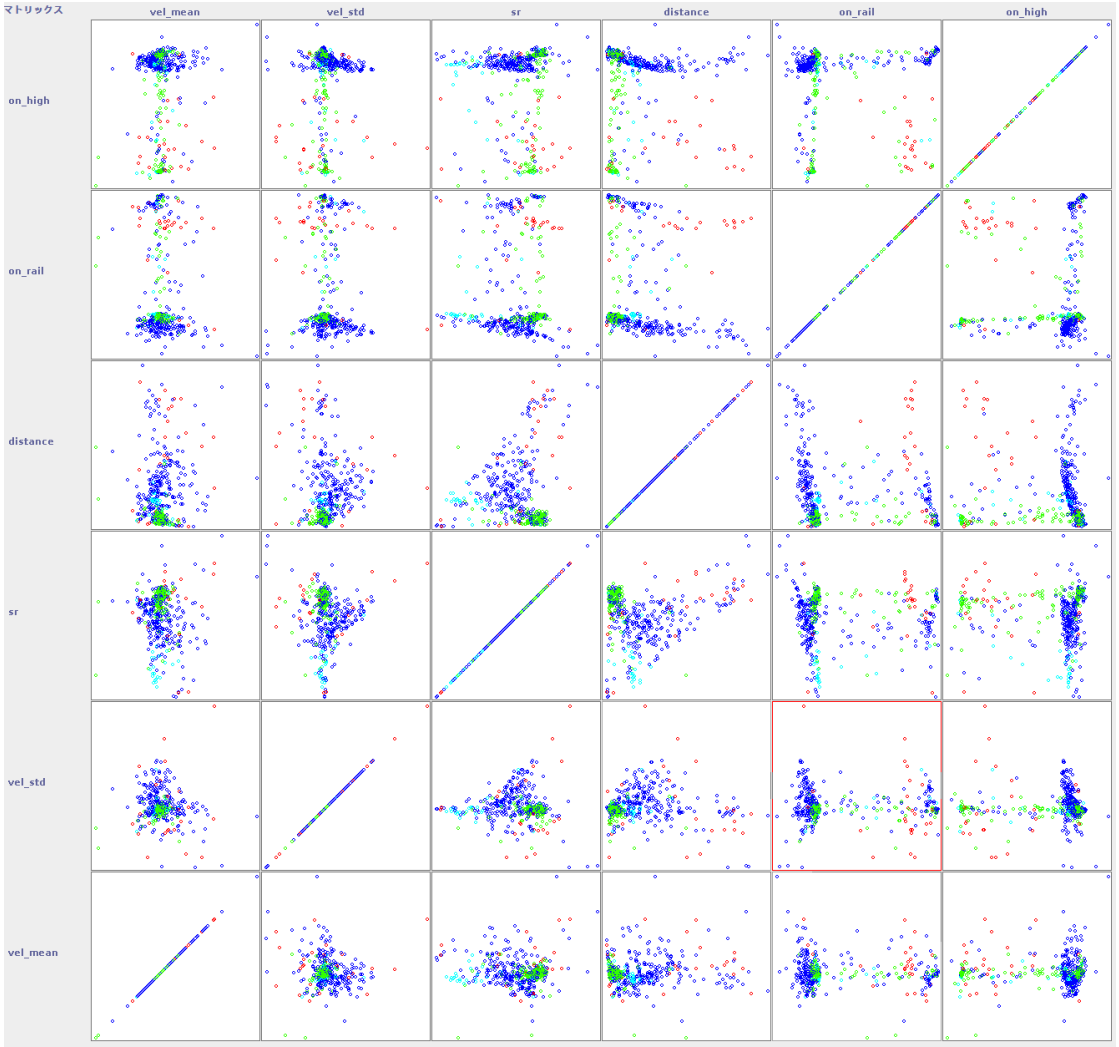

図 4.1: 特徴量の可視化結果.

特徴量の可視化結果. (特徴量:速度の平均 (v_mean)・標準偏差 (v_std), 停止点率 (sr), 距離 (distance), 線路とのマッチ率 (on_rail), 道路とのマッチ率 (on_high)) (モード:car(青), train(赤), bike(水色), walk(緑)).

4.3 Co-Training に関する実験

Co-training を用いて交通移動手段推定を行うための, 調査と実験を行う.

ラベルありデータの集合を L, ラベルなしデータの集合を U とする. Co-training では, 十分なラベルありデータがある場合に分割後の特徴量のみで十分な推定が可能なことと, 分割後の特徴量が条件付き独立であることが仮定として必要であった.

そこで Co-training を用いた交通移動モード推定では, 特徴量 $\mathbf{x}_i$ を物理的特徴量 $\mathbf{x}_i^{(1)}$(速度・加速度の平均・標準偏差・最小・最大, 停止点率, 距離の 10 種) と地理的特徴量 $\mathbf{x}_i^{(2)}$(線路・道路とのマッチ率の 2 種) に分割する. 始めに $L_1 = L_2 = L$ とし, L_1, L_2 を用いて, 各特徴量によってそれぞれ分類器 $f^{(1)}, f^{(2)}$ を構築する. そして, U の各データを $f^{(1)}, f^{(2)}$ のそれぞれで推定し, 各分類器において推定の信頼度が高い上位 k 件を正解とし, 互いの推定結果をそれぞれ L_2, L_1 へ追加する.(アルゴリズム 2)

Algorithm 2: Co-Training

Data: Labeled data: $L = \{(\mathbf{x}_i, y_i)_{i=1}^{l}\}$, unlabeled data $U = \{(\mathbf{x}_j)_{j=l+1}^{l+u}\}$, 学習速度 k

各データは (1) 物理的特徴量と (2) 地理的特徴量の 2 つを持っている
$\mathbf{x}_i = [\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}]$;

initialization $L_1 = L_2 = L$;

while U **が空ではない**. **do**

  L_1, L_2 を用いて, それぞれ分類器 $f^{(1)}, f^{(2)}$ を学習する;

  $f^{(1)}, f^{(2)}$ をそれぞれ独立に U に対し適用する;

  $f^{(1)}$ を用いて推定されたものの中で, 予測の信頼度が高いもの上位 k 件を $(\mathbf{x}, f^{(1)}(\mathbf{x}))$ として, L_2 に加える;

  $f^{(2)}$ を用いて推定されたものの中で, 予測の信頼度が高いもの上位 k 件を $(\mathbf{x}, f^{(2)}(\mathbf{x}))$ として, L_1 に加える;

  L_1, L_2 に新たに加えられたデータを U から取り除く;

end

それぞれの特徴量 (物理的特徴量と地理的特徴量) のみで十分な推定が可能かを調べる実験を行った. 実験では分類器として RF を使用し, 交差数 5 の交差検定で精度を評価した.

4.3.1 実験結果

表 4.2 が実験結果である.

4.3.2 考察

物理的特徴量を用いた推定は, 推定精度が 80% 程度であり分割後の特徴量だけで推定が可能であるが, 地理的特徴量を用いた推定は, 57% ほどであり十分な推定が可

表 4.2: Co-training で分割後の特徴量を用いた推定結果.

—	推定精度	標準偏差
物理的特徴量	80.59	1.37
地理的特徴量	57.04	2.09

能とはならない. そのため, Co-training を推定で用いるには, 追加の地理的特徴量を考える必要がある. 特に現状では道路上または線路上かによる特徴量のみであるので, 図 4.1 の on_rail と on_high のプロットより電車であるか否かの判断はある程度可能であるが, その他のモードの判断は出来ないと考えられる. よって車・自転車・徒歩を分離できる地理的特徴量を発見することが重要となる. 例えば, 歩道上か車道上かによる特徴量を地理的特徴量に追加することで推定精度の向上を図れると予想できるが, 現状の GPS の精度ではそのような特徴量を追加することは難しいだろう.

4.4 Label Propagation (Graph-based semi-supervised) に関する実験

Graph-based 半教師あり学習手法の一種である Label propagation(LP) を用いて交通移動手段推定を行うための, 実験を行う.

LP を用いた推定で重要となるのは, 重みの定義である. 特徴量に関する調査 (図 4.1) で観察できるように, 同じモードの点は特徴量空間中で近い位置に存在するが, モードごとに点の分散は異なっている. そのためユークリッド距離を用いた Radial Basis Function による重みは適切でないと考えられる. そこで重みとしては, k 近傍グラフに基づく重み $w_{ij} = 1$(自身から近い k 番目以内の点), $w_{ij} = 0$(otherwise) を用いる.

実験では k の値として k=5, 10, 15, 20 の 4 つを採用し, 初期のラベル付き軌跡の数は 8 本 (全体のおよそ 1%), 60 本 (全体のおよそ 10%), 236 本 (全体のおよそ 40%) の場合に対して実験を行った. 比較対象としては RF を採用した.

4.4.1 実験結果

実験結果は表 4.3 の通りである.

表 4.3: Labeb Propagation を用いた推定結果.

ラベル付き軌跡 8 本	RF	LP (k=5)	LP (k=10)	LP (k=15)	LP (k=20)
精度	70.81	68.98	71.43	70.88	70.65
標準偏差	3.54	4.64	4.56	4.91	4.68
ラベル付き軌跡 60 本	RF	LP (k=5)	LP (k=10)	LP (k=15)	LP (k=20)
精度	79.74	78.68	80.6	80.73	80.76
標準偏差	1.36	1.33	1.27	1.29	0.93
ラベル付き軌跡 236 本	RF	LP (k=5)	LP (k=10)	LP (k=15)	LP (k=20)
精度	81.91	82.02	82.11	81.76	81.4
標準偏差	0.67	0.30	0.69	0.56	0.65

4.4.2 考察

ラベルありデータの割合が 10% 以下と少ない場合に k=10,15,20 で, 教師あり学習手法である RF より推定精度の平均が高く, 半教師あり学習の効果が出ていると考えられる. これは, 図 4.1 で観察できる通り, モードが同じである点は特徴量空間中で近くに存在するという仮定が正しいからであると考えられる. 一方で, ラベル付きデータの割合が増えると半教師あり学習の効果はなくなることが実験より分かる. 次に k の値に関してだが, k=5 ではラベルありデータの割合が低い場合に推定精度が他の k の値と比べて低いので不適切である. k=10, 15, 20 では平均精度の差は小さいものの, k=10 の場合に全体的に精度が高くなっている. 以上の実験結果と k 近傍グラフの計算量の事も考慮すると, なるべく小さな値であり, かつ推定精度が他の k と比べて低くない k=10 を使用することが適当である.

4.5 まとめ

本章では 1) Self-Training Model, 2) Generative Model, 3) Co-Training, 4) Label Propagation (Graph-based semi-supervised) の 4 種類の半教師あり学習手法をを交通移動モード推定に適用した調査・実験を行った.

4 つの手法の中で, Generative model と Co-training に関しては, 交通移動モード推定への適用は簡単には実現できないであろうことが実験から示された. 一方で, Label propagation と Self-training Model を用いた場合は, 教師あり学習手法を用いるより良い推定精度がでる場合があることを確認した. ただし, Self-training

model はラベルデータの量が少ない場合に著しく推定精度が下がる場合があった. よって今回実験を行った手法の中では Label propagation が最も良い手法と考えられる.

本章の実験の結果を踏まえ, 次章で提案するシステムでは, 推定において Label propagation を拡張した提案手法を用いる.

第 5 章

交通移動モード推定システム

本章では, 前章の結果も踏まえ, 少量のラベル付きデータのみで効率的に交通移動モードを推定するシステムの構築に関して述べる.

5.1 提案システム概要

本システムは入力として GPS 軌跡データ集合とモード集合を入力し, 出力として各点に対しモード集合の中から一つのモードラベルを付与する.

提案システムは図 5.1 で示される通り, 6 つのステップに分けられる. 第 1 のステップでは, GPS により位置情報を収集する. 第 2 のステップでは, GPS データの測定誤差除去のためのスムージングと, 地図情報による特徴量の生成に必要な道路・線路マッチングを行う. 第 3 のステップでは, 軌跡をセグメントへ分割する. 第 4 のステップでは, セグメントごとに特徴量を生成する. 第 5 のステップでは, 特徴量をもとに交通移動モードの学習と予測を半教師あり学習手法を用いて行う. この第 5 のステップでは, 既存の半教師あり学習手法を用いた方法に加え, 交通移動モード推定用に既存手法を拡張した手法の提案も行う. 最後, 第 6 のステップでは各点に対しモードラベルを付与し出力として提示する. 以下では各ステップの詳細を説明する.

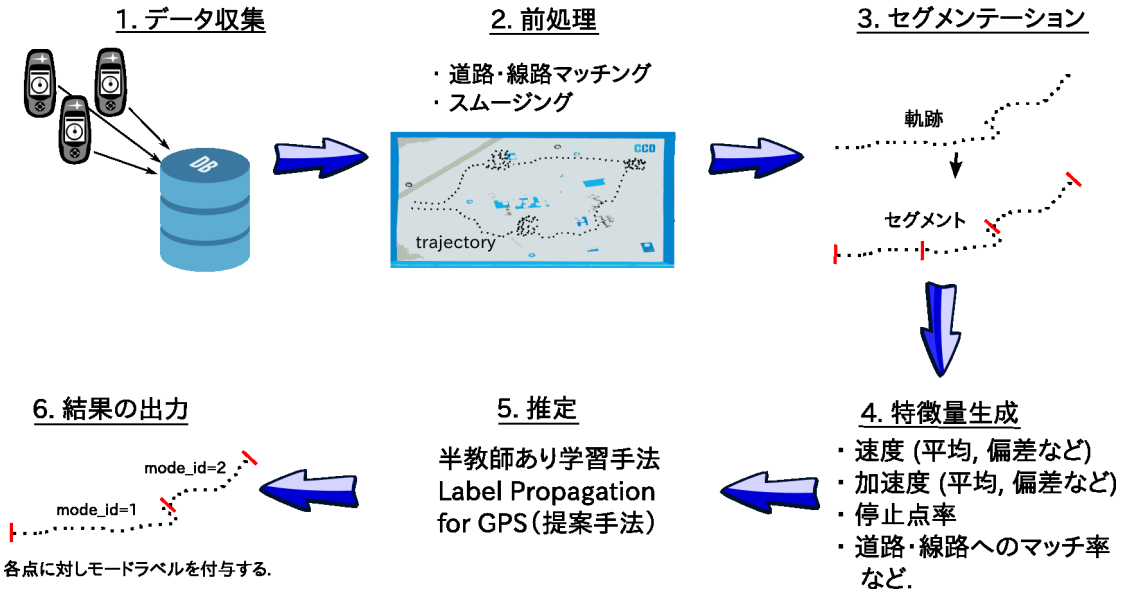

図 5.1: 提案システム概要

5.2 データ収集

位置情報の収集はスマートフォンや専用の GPS 端末などのデバイスを用いて行う. 本システムでは, 後のセグメンテーションや特徴量生成の事を考慮すると, サンプリング間隔は 1 秒から 5 秒程度を想定している.

5.3 前処理

前処理としては 2 つの処理を行う. 第 1 の処理は GPS の測定誤差を除去するためのスムージング処理であり, 第 2 の処理は地図情報による特徴量を生成するために必要となる点と路線のマッチング処理である.

5.3.1 スムージング処理

GPS により収集された位置情報には測定誤差が存在する. その測定誤差を除去するために, カルマンスムージングを行う.

スムージングの説明に入る前に, 座標系の変換に関して簡単に説明する. GPS から取得された位置情報は緯度と経度のペアという地理座標で与えられることが多い. しかし速度や加速度といった物理量を計算する観点からすると, この地理座標系では計算が複雑になり, また直感的にも扱いづらい. そこで始めに, 緯度・経度で表される地理座標を (x,y) で表すことが出来る別の直交座標系に変換する. 本システムでは, 変換にはユニバーサル横メルカトル投影法 (UTM) を使用する. なお投影の詳細に関しては本稿のスコープを外れるため詳しくは [飛田 02] などを参考にされたい.

次にカルマンスムージングに関する説明を行う. なおカルマンスムージングの詳細な計算方法に関しては B を参考にしてもらい, この小節では状態方程式と観測方程式における各変数の具体的な設定に関してのみ説明する.

二次元平面を運動する物体が時刻 t において, 位置 $x(t), y(t)$, 速度 $v_x(t), v_y(t)$, 加速度 $a_x(t), a_y(t)$ を持つとし, 状態変数 $\mathbf{x}_t$ を,

$$\mathbf{x}_t = (x(t), y(t), v_x(t), v_y(t), a_x(t), a_y(t))^T$$

とする.

またニュートン方程式により,

$$
\begin{aligned}
x(t+1) &= x(t) + v_x(t)\Delta t + 0.5a_x(t)\Delta t^2 \\
y(t+1) &= y(t) + v_y(t)\Delta t + 0.5a_y(t)\Delta t^2 \\
v_x(t+1) &= v_x(t) + a_x(t)\Delta t \\
v_y(t+1) &= v_y(t) + a_y(t)\Delta t
\end{aligned}
$$

であるので, 状態遷移行列 $\mathbf{A}$ は,

$$
\mathbf{A} = \begin{pmatrix}
1 & 0 & \Delta t & 0 & 0.5\Delta t^2 & 0 \\
0 & 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 \\
0 & 0 & 1 & 0 & \Delta t & 0 \\
0 & 0 & 0 & 1 & 0 & \Delta t \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix} \tag{5.1}
$$

となる. なお Δt は時刻 $t+1$ と t の時間間隔であり, 具体的には GPS のサンプリング間隔である.

また観測値としては, 緯度・経度のみが観測可能であるため,

$$
\mathbf{y}_t = (x(t), y(t))^T \tag{5.2}
$$

となり, 観測行列 $\mathbf{C}$ は,

$$
\mathbf{C} = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0
\end{pmatrix} \tag{5.3}
$$

となる.

本システムでは以上に挙げた変数を使用しスムージング処理を行う. またスムージングの際に得られた各点における速度 $v(t) = \sqrt{v_x(t)^2 + v_y(t)^2}$ を点の速度として使用する.

5.3.2 路線マッチング処理

路線マッチング処理では, 特徴量生成の際に必要となる情報として, GPS の点が路線 (道路と線路) に乗っているか否かを判定する処理を行う.

図 5.2 は中国北京市内の東西約 25km, 南北約 30km の交通網を表しており, 図 5.3 はその一部を拡大したものである. 図で分かる通り, 道路 (図では緑線) や線路 (図では青線) の情報はラインで表されることが一般的である.

路線マッチング処理では, 路線上に GPS の点があるか否かの判定を行うが, 路線を表すラインは, 路線の中央に 1 本であったり, 路線の両端に 1 本ずつといった表現をされている場合が多く, 路線マッチング処理のためには路線の幅を考慮する必要が

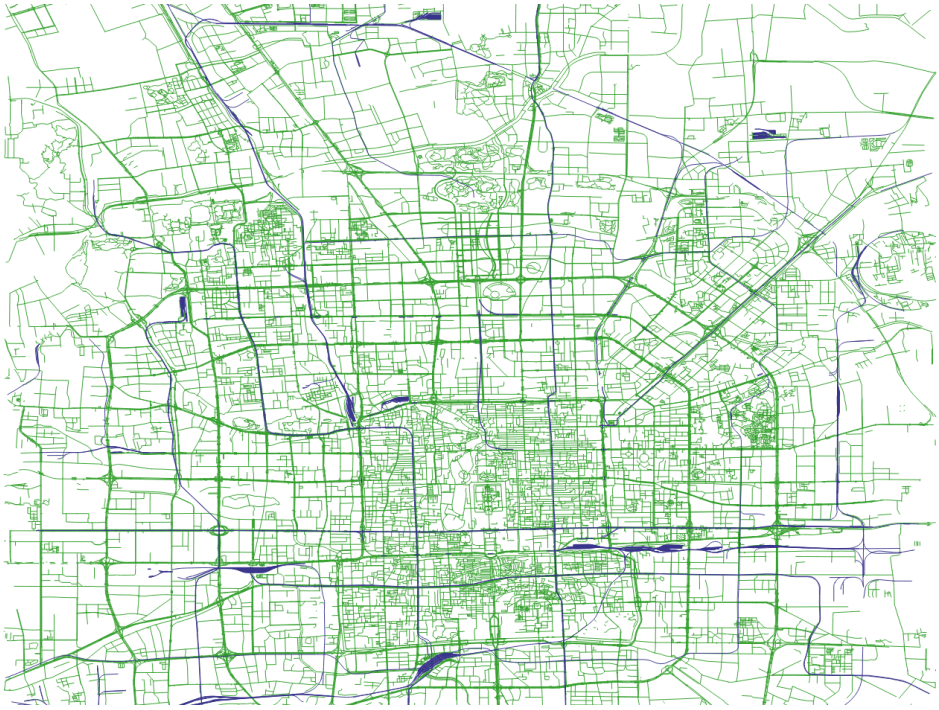

図 5.2: 交通網 (広域).

青ラインが線路, 緑ラインが道路. (©OpenStreetMap contributors, CC BY-SA)

ある. また GPS の点には誤差があるため, その誤差も考慮に入れる必要がある. なお本システムでは, 地図情報に OpenStreetMap[*1] を使用しているので, ラインは路線の中央を表している. そこで, まずライン (図 5.4 左) に対しバッファを与え (図 5.4 右), ポリゴンを形成する. そして, そのポリゴン内に GPS の点が存在すれば, 道路又は線路上に点があるものと判定する. なお本システムではバッファ幅は道路・線路ともに 30m とした.

5.4 セグメンテーション

セグメンテーションのステップでは, 軌跡を複数のセグメントへ分割する (図.5.5).

本システムでは, 軌跡 $T = \langle p_0, p_1, ..., p_n \rangle$ をセグメント $S = \langle p_i, p_{i+1}, ..., p_{i+m} \rangle (1 \leq i < i + m \leq n)$ へ分割する方法は 2 種類あり, 1) 時間分割, 2) 距離分割となる.

時間分割は時間幅 τ により分割する方法である. 各点 p_i の時刻を $p_i.t$ と表すと, 時間分割とは $T = \langle p_0, p_1, ..., p_n \rangle$ をセグメント $S = \langle p_i, p_{i+1}, ..., p_{i+m} \rangle (1 \leq i <$

[*1] http://www.openstreetmap.org/

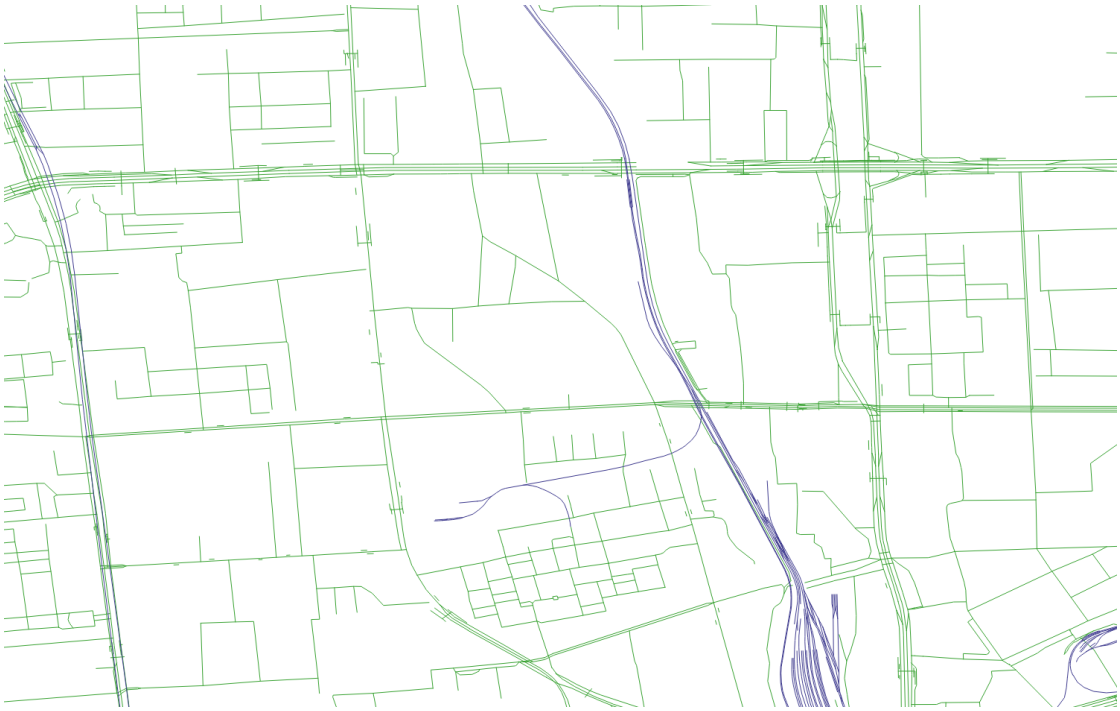

図 5.3: 交通網 (狭域).

青ラインが線路, 緑ラインが道路. (©OpenStreetMap contributors, CC BY-SA)

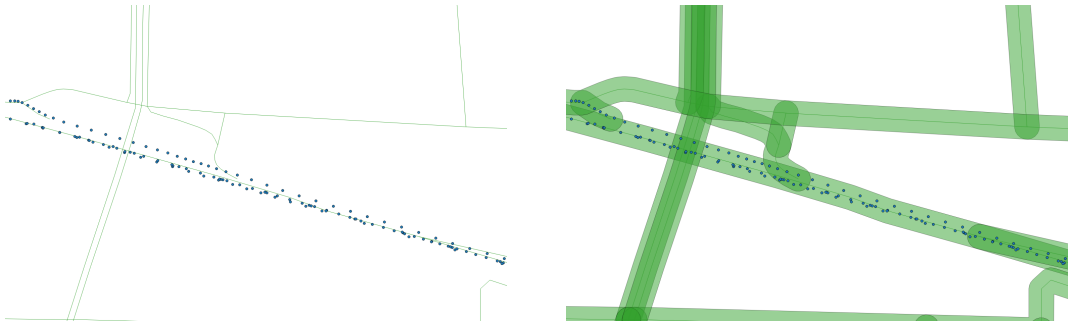

図 5.4: 道路・線路ラインとバッファ.

左図の緑ラインは道路, 青点は GPS 点. 右図が左図のラインに対しバッファを生成させたもの. (Map data ©OpenStreetMap, CC BY-SA)

図 5.5: セグメンテーション.

$i+m \leq n), (p_{i+m}.t - p_i.t \leq \tau)$ へ分割することである. ただし各点は 1 つのセグメントへ属するものとするため, $T = \langle S_0, ..., S_{end} \rangle$ とする.

距離分割とは距離幅 d により分割する方法である. 2 点 p_i, p_j の距離を $Dist(p_i, p_j)$ と表すと, 距離分割とは $T = \langle p_0, p_1, ..., p_n \rangle$ をセグメント $S = \langle p_i, p_{i+1}, ..., p_{i+m} \rangle (1 \leq i < i+m \leq n), (\sum_{j=i}^{j=i+m-1} Dist(p_j, p_{j+1}) \leq d)$ へ分割することである. ただし各点は 1 つのセグメントへ属するものとするため, $T = \langle S_0, ..., S_{end} \rangle$ とする.

5.5 特徴量生成

特徴量生成のステップでは, セグメント内の各点から得られる情報をもとにして, 各セグメントに対して特徴量を生成する. 更に生成された特徴量は, 特徴量間の分散の違いを取り除くため白色化 (whitening) を行い, 平均が 0 で共分散行列が単位行列となるように変換する (詳しい計算方法は [Bis06] を参照). 以下ではセグメントは $S = \langle p_0, p_1, ..., p_{N-1} \rangle$ として説明する. 表 5.1 は, 本システムで扱う特徴量の一覧である.

表 5.1: 特徴量一覧.

速度の平均値	速度の標準偏差	速度の最大	速度の最小
加速度の平均値	加速度の標準偏差	加速度の最大	加速度の最小
停止点率	鉄道とのマッチ率	道路とのマッチ率	所要距離・時間

5.5.1 速度の平均・標準偏差・最大・最小

前処理のカルマン・スムージングを行った際に, 各点には速度の情報が既に付与されている. S 内の各点 p_i の速度を $p_i.v$ と表記すると, 速度の平均, 標準偏差, 最大, 最小は次の通りとなる.

$$速度の平均\ \mu_{vel} = \frac{1}{N}\sum_{i=0}^{i=N-1} p_i.v \tag{5.4}$$

$$速度の標準偏差\ \sigma_{vel} = \sqrt{\frac{1}{N}\sum_{i=0}^{i=N-1}(p_i.v - \mu_{vel})^2} \tag{5.5}$$

$$速度の最大\ v_{max} = max(p_0.v, ..., p_{N-1}.v) \tag{5.6}$$

$$速度の最小\ v_{min} = min(p_0.v, ..., p_{N-1}.v) \tag{5.7}$$

5.5.2 加速度の平均・標準偏差・最大・最小

点 $p_i(i = 1, ..., N-1)$ の加速度を $p_i.a$, 時刻を $p_i.t$ と表記し, $p_i.a = \frac{p_i.v - p_{i-1}.v}{p_i.t - p_{i-1}.t}$ とすると, 加速度の平均, 標準偏差, 最大, 最小は次の通りとなる.

$$加速度の平均\ \mu_{acc} = \frac{1}{N-1}\sum_{i=1}^{i=N-1} p_i.a \tag{5.8}$$

$$加速度の標準偏差\ \sigma_{acc} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{i=N-1}(p_i.a - \mu_{acc})^2} \tag{5.9}$$

$$加速度の最大\ a_{max} = max(p_1.a, ..., p_{N-1}.a) \tag{5.10}$$

$$加速度の最小\ a_{min} = min(p_1.a, ..., p_{N-1}.a) \tag{5.11}$$

5.5.3 停止点率

停止点率 (Stop Rate)[ZCL+10] は, セグメント内の点の速度がある閾値 V_s 以下になった点の総数を $|PS|$ とすると, 次の通りとなる.

$$停止点率\ SR = \frac{|PS|}{N} \tag{5.12}$$

5.5.4 所要距離・所要時間

所要距離と所要時間はセグメンテーションの方法によって一方のみが使用可能な特徴量である. セグメンテーションの方法が時間分割の場合は所要距離を, 距離分割の場合は所要時間を用いる. $dist(p, q)$ を点 p, q 間の直線距離とする.

$$\text{所要距離 } Dist = dist(p_0, p_{N-1}) \tag{5.13}$$

$$\text{所要時間 } Time = p_{N-1}.t - p_0.t \tag{5.14}$$

5.5.5 線路・道路とのマッチ率

路線マッチング処理の結果, 点 p_i が, 線路上に存在すれば $p_i.on\_railway = True$, 道路上に存在すれば $p_i.on\_highway = True$ とする. セグメント内の点で $p_i.on\_railway = True$, $p_i.on\_highway = True$ となっている点の総数をそれぞれ N_{rail}, N_{high} とすると, 線路・道路とのマッチ率はそれぞれ次の通りとなる.

$$\text{線路マッチ率 } MR_{rail} = \frac{N_{rail}}{N} \tag{5.15}$$

$$\text{道路マッチ率 } MR_{high} = \frac{N_{high}}{N} \tag{5.16}$$

5.6 推定 (提案手法)

推定のステップではセグメントの特徴量を用いて, 各セグメントがどのモードに属するかを推定する.

本システムでは推定方法として, 4 章の半教師あり学習手法の実験の中で最も良い精度を出した Label Propagation(LP) を, 交通移動モード推定用に拡張した推定方法 Label Propagation for GPS(以下 LP_GPS) を提案し使用する.

LP では, 特徴量間の近さのみに基づいて重みを計算しラベルの推定を行った. しかし現実の人の移動を考えると, 交通移動モードはセグメントの前後で頻繁に変わることはない (2.3.1 章 図 2.1). つまり, 多くの場合セグメントの前後ではモードは同じである可能性が高い. そこで LP_GPS では, 特徴量間の近さのみに基づくのではなく, セグメント同士が近いかという情報も同時に取り入れる. 具体的には, LP において隣り合うセグメント間 (ノード間) のエッジの重み w_{ij} を釣り上げる操作を行う. 特徴量だけでなくセグメント同士の近さ情報を重みに含めることで, 決定境界付近での判断を特徴量だけでなくセグメントの近さに基づいて行うことが出来る.

$\mathbf{x}^{\{t,s\}}$ を軌跡 t の s 番目のセグメントによる特徴量とし, $\mathbf{X} = \{x_1^{\{1,1\}}, x_2^{\{1,2\}}, ..., x_N^{\{Trj,T\_end\}}\}$ とする. なお N はセグメントの総数であり, Trj は軌跡の総数, T_end は軌跡 Trj の最後のセグメントを表す. 点 $\mathbf{x}_i$ と点 $\mathbf{x}_j$ の近さ (類似度) を表す重み w_{ij} による

$N \times N$ 行列を $\mathbf{W} = \left[w_{ij}\right]$ とする. LP_GPS では, 特徴量による近さだけでなくセグメントの近さも考慮するために, この $\mathbf{W}$ を次のように $\mathbf{W}'$ におきかえる.

$$\mathbf{W}' = (1-\beta)\mathbf{W} + \beta\mathbf{A} \quad (0 \leqq \beta \leqq 1) \tag{5.17}$$

$\mathbf{A}$ はセグメント i とセグメント $i+1$ が隣あっていれば, つまり $\mathbf{x}_i^{\{t,s\}}$かつ $\mathbf{x}_{i+1}^{\{t,s+1\}}$ であれば $a_{i+1,i} = a_{i,i+1} = 1$, そうでなければ 0 であり, その他, 対角成分は 1, それら以外は 0 である. また β は近接セグメントの影響をどの程度考慮するかの調整パラメータであり, β が 1 に近いほど近接セグメントの影響を強く考慮する.

以後は, この新たな $\mathbf{W}' = \left[w'_{ij}\right]$ を用いて遷移確率行列 $\mathbf{T}$ を計算し, LP と同様の計算によってラベルなしデータのラベルを予測する.

第 6 章

提案システムの評価実験

本章では提案システムの各ステップに関する評価実験を行う.

6.1 実験環境

本実験で使用したライブラリーやシステムは表.6.1 の通りである.

表 6.1: 実験環境 (言語・ライブラリー・システム).

言語	Python2.7
ライブラリー	Numpy, Scipy, Scikit-learn, Pykalman, pyproj
データベース	PostgreSQL, SpatiaLite
地理情報システム	QGIS
地図情報	Open Street Map

6.2 評価方法

評価実験で使用するデータセットは Microsoft Research の GeoLife GPS Trajectories [*1] [ZLC+08, ZZXM09, ZCXM09] (以下 GeoLife) から生成したデータセット (表 A.4) である. データセット中の, 点の総数は 324,875 点であり, モード数は 4 である. 各モードにおける点の総数は walk が 103,849 点, train が 14,809 点, bike が 58,983 点, car が 147,235 点であり, 軌跡の本数の合計は 588 本である. なおデータセット生成の詳細に関しては付録 A を参照.

表 6.2: 実験データセット.

—	walk	train	bike	car	total
ポイント数	103,849	14,809	58,983	147,235	324,876
データの割合	31.96%	4.56%	18.15%	45.32%	100.00%
軌跡数の合計	588	–	–	–	–

実験における評価は次のように行う. データセット全体を $D(|D| = 324,876)$, ラベル付きデータを D_l, ラベルなしデータを D_u とする (ただし $D = D_l + D_u$). 特に

[*1] http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/

断りがない限り, 各実験はラベル付きデータをランダムに変更し 10 回試行し, 精度の平均と標準偏差を求める. 精度とは,

$$\text{精度} = \frac{D_u\text{の中で正しく推定された点の数}}{D_u\text{の点の総数}} \times 100\ [\%] \tag{6.1}$$

である. また, ラベル付きデータ D_l の生成は, 軌跡単位で行う. つまり, 軌跡の総数 588 本の中で何本の軌跡にラベルが付いているかを変更しながら実験は行われる. そのため, 例えばラベル付き軌跡の数が 8 本の場合の実験であっても, 各試行ごとのラベル付きデータの個数 $|D_l|$ は異なっている. ラベル付き軌跡の選択方法は, ラベル付き軌跡の本数が 20 本以下の場合は, 各モードの軌跡の割合が等しくなるよう選択し, 20 本より多い場合は, 最低でも各モードの軌跡が 5 本ずつ存在するように選択する.

6.3 特徴量に関する実験

本節では, 5.5 章で説明したどの特徴量が推定に効果があるのかを実験で確認する.

特徴量が N 個ある場合には, $2^N - 1$ パターンの特徴量の選択の仕方があるが, 特徴量数が増えるに従いこれら全てを確認することは計算量的に困難となる. そこで特徴の検証にあたっては Forward stepwise selection[GE03] を使用する. この手法は各ステップで有効であると判断された特徴量を一つずつ追加していくことで, 有効な特徴量選択を検証する方法である.

検証では学習器として教師あり学習手法のランダムフォレストを使用し, ラベル付き軌跡の本数は 116 本 (およそ全体の 20%) とした. またセグメンテーションの方法は時間分割・距離分割の両方を試し, 時間分割の時間幅は 60 秒, 距離分割の距離幅は 100 m とする.

6.3.1 実験結果

表 6.3 から表 6.12 は, セグメンテーション方式として時間分割を用い推定した結果であり, 表 6.13 から表 6.22 は, セグメンテーション方式として距離分割を用い推定した結果である. 表内の固定特徴量は, それ以前に有効と判断された特徴量を表していて, 追加特徴量は固定特徴量に対し追加する特徴量を表している. また, *印は各ステップにおいて最も精度が高い値を表す.

6.3.2 考察

実験の結果, どちらの方式においても停止点率は効果的な特徴量である. また, 地理的特徴量の線路とのマッチ率・道路とのマッチ率や速度の平均値も効果的な特徴量であることがわかる. 時間分割をした場合は, それらに加え速度の加速度の平均値・速度の最小なども有効である. 距離分割をした場合は, それらに加え所要時間・加速度の最大値なども有効である.

表 6.3: 時間分割 (ステップ 1).

固定特徴量	なし	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	64.16	0.52
vel_std	60.25	1.17
vel_min	49.70	0.98
vel_max	65.59	1.48
acc_mean	53.72	0.79
acc_std	55.14	2.10
acc_min	51.31	1.37
acc_max	50.44	1.69
sr	72.48*	1.17
dist	61.55	1.49
mr_rail	46.42	0.21
mr_high	55.35	1.65

表 6.4: 時間分割 (ステップ 2).

固定特徴量	sr	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	72.07	0.54
vel_std	70.59	1.01
vel_min	67.75	0.87
vel_max	70.42	1.90
acc_mean	67.62	0.68
acc_std	67.54	1.41
acc_min	66.02	0.98
acc_max	66.09	1.02
dist	70.63	1.21
mr_rail	73.11	1.45
mr_high	74.17*	0.84

表 6.5: 時間分割 (ステップ 3).

固定特徴量	sr, mr_high	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	75.72*	0.72
vel_std	74.48	1.11
vel_min	72.05	0.79
vel_max	74.90	1.55
acc_mean	72.38	0.50
acc_std	71.10	1.48
acc_min	69.92	0.29
acc_max	70.52	0.81
dist	74.49	1.48
mr_rail	75.10	0.85

表 6.6: 時間分割 (ステップ 4).

固定特徴量	sr, mr_high, vel_mean	
追加特徴量	精度の平均	精度の標準偏差
vel_std	79.92	1.62
vel_min	79.82	1.44
vel_max	79.33	1.37
acc_mean	80.17*	1.23
acc_std	79.40	1.32
acc_min	79.46	1.82
acc_max	79.87	1.10
dist	79.42	1.72
mr_rail	77.48	1.16

表 6.7: 時間分割 (ステップ 5).

固定特徴量	sr, mr_high, vel_mean, acc_mean	
追加特徴量	精度の平均	精度の標準偏差
vel_std	80.77	1.18
vel_min	81.10*	1.32
vel_max	80.53	1.31
acc_std	81.00	1.50
acc_min	80.98	1.54
acc_max	81.04	1.30
dist	80.56	1.44
mr_rail	80.87	1.28

表 6.8: 時間分割 (ステップ 6).

固定特徴量	sr, mr_high, vel_mean, acc_mean, vel_min	
追加特徴量	精度の平均	精度の標準偏差
vel_std	81.13	1.22
vel_max	80.98	1.43
acc_std	81.35	1.33
acc_min	81.40	1.33
acc_max	81.43	1.27
dist	81.59	1.55
mr_rail	81.80*	1.21

表 6.9: 時間分割 (ステップ 7).

固定特徴量	sr, mr_high, vel_mean, acc_mean, vel_min, mr_rail	
追加特徴量	精度の平均	精度の標準偏差
vel_std	82.07	1.22
vel_max	81.84	1.08
acc_std	81.90	1.18
acc_min	81.99	1.18
acc_max	82.05*	1.26
dist	81.94	1.37

表 6.10: 時間分割 (ステップ 8).

固定特徴量	sr, mr_high, vel_mean, acc_mean, vel_min, mr_rail, acc_max, acc_max	
追加特徴量	精度の平均	精度の標準偏差
vel_std	82.37*	1.27
vel_max	82.23	1.10
acc_std	82.11	1.34
acc_min	82.19	1.35
dist	82.27	1.04

表 6.11: 時間分割 (ステップ 9).

固定特徴量	sr, mr_high, vel_mean, acc_mean, vel_min, mr_rail, acc_max, acc_max, vel_std	
追加特徴量	精度の平均	精度の標準偏差
vel_max	82.64*	1.39
acc_std	82.34	1.38
acc_min	82.36	1.36
dist	82.13	1.37

表 6.12: 時間分割 (ステップ 10).

固定特徴量	sr, mr_high, vel_mean, acc_mean, vel_min, mr_rail, acc_max, acc_max, vel_std, vel_max	
追加特徴量	精度の平均	精度の標準偏差
acc_std	82.42	1.41
acc_min	82.56	1.20
dist	82.62*	1.22
使用特徴量	精度の平均	精度の標準偏差
全て	82.69	1.37

表 6.13: 距離分割 (ステップ 1).

固定特徴量	なし	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	56.54	1.34
vel_std	41.04	2.84
vel_min	50.87	0.92
vel_max	62.02	1.99
acc_mean	47.50	2.11
acc_std	39.65	2.12
acc_min	38.48	1.92
acc_max	39.32	1.42
sr	67.17*	1.09
time	60.27	1.14
mr_rail	45.53	0.90
mr_high	50.65	1.62

表 6.14: 距離分割 (ステップ 2).

固定特徴量	sr	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	67.39	1.36
vel_std	62.83	1.10
vel_min	66.26	1.33
vel_max	68.62	1.32
acc_mean	62.56	1.50
acc_std	61.08	1.94
acc_min	61.07	1.40
acc_max	60.73	1.10
time	68.46	1.47
mr_rail	66.68	0.94
mr_high	68.63*	0.89

表 6.15: 距離分割 (ステップ 3).

固定特徴量	sr, mr_high	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	71.15	1.46
vel_std	67.68	1.14
vel_min	69.31	1.27
vel_max	72.21	0.70
acc_mean	68.12	0.98
acc_std	65.44	0.88
acc_min	65.89	0.84
acc_max	65.53	1.44
time	72.75*	1.46
mr_rail	68.67	1.45

表 6.16: 距離分割 (ステップ 4).

固定特徴量	sr, mr_high, time	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	75.20	1.42
vel_std	75.26	0.93
vel_min	74.10	1.30
vel_max	75.67*	0.87
acc_mean	74.40	0.83
acc_std	73.45	0.83
acc_min	74.04	0.94
acc_max	73.48	1.08
mr_rail	73.44	1.30

表 6.17: 距離分割 (ステップ 5).

固定特徴量	sr, mr_high, time, v_max	
追加特徴量	精度の平均	精度の標準偏差
vel_mean	78.02*	0.64
vel_std	76.76	0.97
vel_min	77.13	1.19
acc_mean	77.03	1.18
acc_std	77.45	1.04
acc_min	77.60	0.94
acc_max	78.00	0.64
mr_rail	77.10	0.98

表 6.18: 距離分割 (ステップ 6).

固定特徴量	sr, mr_high, time, vel_max, vel_mean	
追加特徴量	精度の平均	精度の標準偏差
vel_std	78.06	0.82
vel_min	77.74	1.35
acc_mean	77.50	1.40
acc_std	77.43	1.47
acc_min	77.93	0.93
acc_max	77.85	0.94
mr_rail	78.28*	1.00

表 6.19: 距離分割 (ステップ 7).

固定特徴量	sr, mr_high, time, vel_max, vel_mean, mr_rail	
追加特徴量	精度の平均	精度の標準偏差
vel_std	79.07	1.28
vel_min	79.00	0.84
acc_mean	78.83	0.70
acc_std	79.20	1.14
acc_min	78.09	1.09
acc_max	79.42*	1.03

表 6.20: 距離分割 (ステップ 8).

固定特徴量	sr, mr_high, time, vel_max, vel_mean, mr_rail, acc_max	
追加特徴量	精度の平均	精度の標準偏差
vel_std	79.46*	0.69
vel_min	79.19	1.03
acc_mean	79.23	1.07
acc_std	78.70	1.45
acc_min	79.42	1.24

表 6.21: 距離分割 (ステップ 9).

固定特徴量	sr, mr_high, time, vel_max, vel_mean, mr_rail, acc_max, vel_std	
追加特徴量	精度の平均	精度の標準偏差
vel_min	79.97*	0.98
acc_mean	79.08	0.93
acc_std	79.93	1.09
acc_min	79.56	0.78

表 6.22: 距離分割 (ステップ 10).

固定特徴量	sr, mr_high, time, vel_max, vel_mean, mr_rail, acc_max, vel_std, vel_min	
追加特徴量	精度の平均	精度の標準偏差
acc_mean	80.18	1.50
acc_std	80.17	1.24
acc_min	79.61	0.94
使用特徴量	精度の平均	精度の標準偏差
全て	80.37	0.90

6.4 セグメンテーションに関する実験

時間分割と距離分割の 2 種類のセグメンテーション方法に関する実験を行う. 時間分割の方法として時間幅 10 秒, 30 秒, 60 秒を, 距離分割の方法として距離幅 50 m, 100 m, 150 m の合計 6 種類に対して, ラベルの量が多い場合 (割合がおよそ 20%) と少ない場合 (割合がおよそ 5%) の 2 通りに対し実験を行った. 推定手法は RF と LP の 2 種類を使用し, 特徴量は速度の平均・標準偏差, 停止点率, 距離/時間, 線路・道路とのマッチ率の 6 種類を採用した.

6.4.1 実験結果

表 6.23 はセグメンテーション方式として時間分割を採用した場合の実験結果であり, 表 6.24 はセグメンテーション方式として距離分割を採用した場合の実験結果である. 表の値は平均精度を表していて, ±の部分は標準偏差である.

表 6.23: 実験結果：セグメンテーション (時間分割).

ラベル付き軌跡の数	10 秒	30 秒	60 秒
28 (5%,RF)	67.33 ± 1.48	72.75 ± 1.56	77.59 ± 2.30
28 (5%,LP)	67.05 ± 1.44	74.01 ± 1.56	78.01 ± 2.44
296 (50%,RF)	72.32 ± 1.20	77.29 ± 1.22	81.12 ± 2.70
296 (50%,LP)	71.91 ± 1.56	78.39 ± 1.05	81.19 ± 1.52

表 6.24: 実験結果：セグメンテーション (距離分割).

ラベル付き軌跡の数	50[m]	100[m]	150[m]
28 (5%,RF)	70.85 ± 1.67	75.87 ± 1.47	77.12 ± 2.42
28 (5%,LP)	68.88 ± 2.24	75.84 ± 1.60	76.72 ± 2.84
296 (50%,RF)	76.18 ± 1.77	78.79 ± 1.12	80.81 ± 1.46
296 (50%,LP)	75.01 ± 2.23	79.08 ± 1.14	80.52 ± 1.70

6.4.2 考察

時間幅 60 s の時間分割が RF, LP 共に最も平均精度が高い. また割時間分割・距離分割共にセグメントの長さが伸びるほど精度が向上している. ラベルの量が増えると推定精度は向上するが, セグメントの長さによる精度の伸びはラベルの量が少ない場合の方が大きいことがわかる. 時間分割・距離分割共に, セグメントの長さが短い場合は RF の方が平均精度が高く, セグメントの長さが長くなるにつれて LP の方が平均精度が高くまたは同じ程度になっている. よって, 特に LP を用いる場合は, セグメントがある程度の長さを持たないと, 各モードの違いを有効に表す特徴量が生成できないことが実験よりわかる.

6.5 推定 (提案手法) に関する実験

提案手法である LP_GPS に関する実験を行う.

以下の実験では, 特徴量は速度の平均・標準偏差, 停止点率, 距離/時間, 線路・道路とのマッチ率の 6 種類を採用し, セグメンテーション方法は時間分割の時間幅 60 s を採用する. LP_GPS の重み $\mathbf{W}$ には, k 近傍グラフによる重みを採用し, 章の実験結果より k=10 とした.

6.5.1 パラメータ β に関する実験

始めに LP_GPS のパラメータ β に関する実験を行う. ラベル付き軌跡の数を全体の約 10% となる 60 本に固定し, β の値を 0.1 から 0.9 まで 0.1 刻みで変化させ実験を行った.

表 6.25 が β に関する実験の結果である.

実験の結果, β の値が 0.1 または 0.2 の場合に推定精度が良い.

表 6.25: LP_GPS のパラメータ β に関する実験.

β	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
平均精度	81.47	81.50	80.84	79.93	78.62	77.49	77.27	78.53	81.11

6.5.2 LP_GPS の推定精度に関する実験

前節の結果を踏まえて, $\beta = 0.2$ を採用して LP_GPS の推定精度に関する実験を行った. 比較対象は RF と LP の 2 つを採用した. RF の調整パラメータは, 特徴選択の規準と決定木の個数である. 実験では規準としてジニ係数とエントロピーの 2 種, 決定木の個数として 5, 10, 15, 20 の 4 種の合計 8 パターンのパラメータを試した. 規準としてジニ係数, 決定木の個数は 15 個の場合が最も推定精度が高かったので, その結果を比較規準として採用した. LP のパラメータは LP_GPS と同様に, k 近傍グラフによる重みを k=10 とし採用した.

表 6.26 が実験結果である. 1 番左の行はラベル付き軌跡の数を表していて, 括弧の中はデータ全体に対するラベルのおよその割合を表している. それ以外の 3 行は, それぞれ推定手法として RF, LP, LP_GPS を用いた場合の平均推定精度を表していて, ±の部分は標準偏差を表している.

表 6.26: 提案手法 LP_GPS に関する実験結果.

# labeled trajectories	RF	LP	LP_GPS(β =0.2)
8 (1%)	73.81 ± 4.219	74.28 ± 4.221	75.3 ± 6.807
28 (5%)	78.145 ± 1.462	78.905 ± 1.823	80.95 ± 1.324
60 (10%)	79.445 ± 0.836	80.5 ± 0.953	80.97 ± 1.472
116 (20%)	80.675 ± 0.749	81.255 ± 0.706	81.285 ± 1.519
176 (30%)	81.365 ± 0.517	81.585 ± 0.897	81.195 ± 1.315
236 (40%)	81.78 ± 1.035	81.675 ± 0.872	81.4 ± 1.089
296 (50%)	81.445 ± 1.233	81.325 ± 1.211	81.12 ± 1.539

6.5.3 考察

ラベルの割合が 30% 以上の場合は教師あり学習との差はないことが実験より分かるが, ラベルの割合が 10% 以下の場合には半教師あり学習を用いた推定は, 教師あり学習を用いた推定に対し平均して 2%〜3% 程度の精度の向上がみられるため, 有効であることが確認できる. LP と LP_GPS を比較するとラベルの割合が少ない場合に, 平均して LP_GPS の方が高い精度を示している. ただし, LP_GPS は他の 2 手法に比べてラベルの割合が少ない場合には, 精度の標準偏差が大きい. また, いずれの

手法を用いた場合でも, 推定精度は 81% 台で頭打ちとなっているが, 半教師あり学習手法の方が少ないラベルの量頭打ちとなっていることがわかる. よってラベルの割合が少ない場合には, 平均すると LP_GPS は RF と LP よりも有効な手法である.

6.6 推定精度改善のための考察事項

この節では LP_GPS の出力結果を具体的に観察し, 精度向上のためにはどのような点に注意し, そして改善するべきかを考察する. 考察項目は次の 2 点である,

1. ラベルの付け方に関して.
2. 誤推定に関して.

6.6.1 ラベルの付け方に関して

実験結果 (表 6.26) より LP_GPS は, ラベルの割合が少ない場合 (10% 以下の場合) には他の 2 手法に比べ精度の標準偏差が大きいことが分かる. これは初期のラベルの与え方次第で, 精度が大きく左右されることを示している. そこで, どのような軌跡に対し初期ラベルを与えれば高い推定精度をだせるかに関して考察を行う.

図 6.1 と図 6.2 は, RF での推定精度に大きな差はみられないが, LP_GPS での推定精度に差がでた 2 つの実験の, ラベル付きデータの特徴量の分布をプロットした図である. ラベル付き軌跡の本数は 8 本 (全体のおよそ 1%) とし, 図 6.1 の推定精度は RF で 72.0%, LP_GPS で 79.8% であり, 図 6.2 の推定精度は RF で 73.1%, LP_GPS で 69.1% である.

2 つの図で注目すべき所は, 特徴量の dist(距離) と sr(停止点率) をプロットしたマスである. LP_GPS での推定精度が高かった図 6.1 では, walk/bike/car の 3 つのモードが図 6.2 に比べて, 明確に分離していることが観察できる. この傾向は, LP_GPS での推定精度が高かった多くの実験で観察された. よってラベル付きデータの特徴量をプロットし, dist と sr によって walk/bike/car が明確に分離しているかを観察することは, ラベル付きデータ作成の一つの指針となり得る.

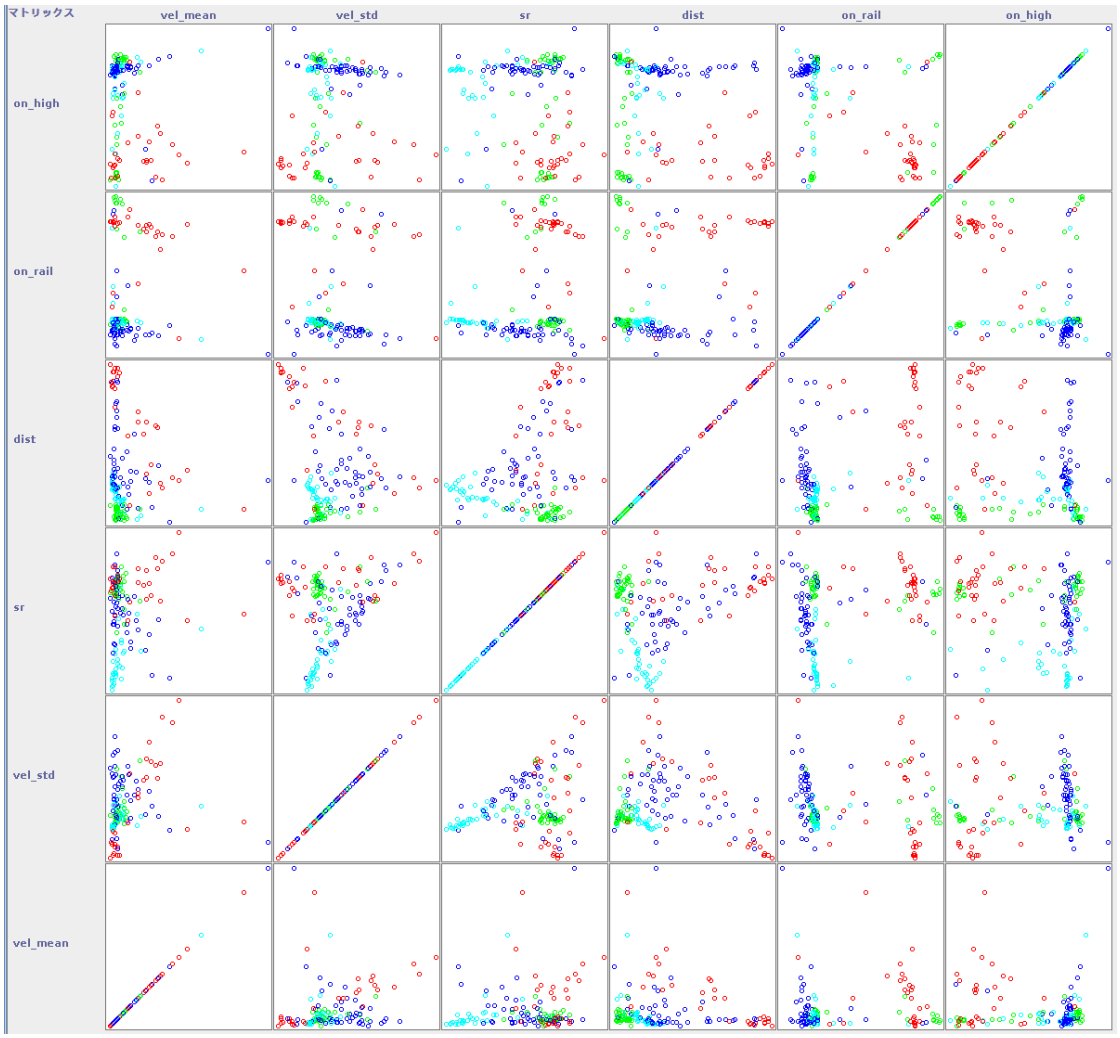

図 6.1: 適切なラベル付きデータの分布.

特徴量:速度の平均 (vel_mean)・標準偏差 (vel_std), 停止点率 (sr), 距離 (dist), 線路とのマッチ率 (on_rail), 道路とのマッチ率 (on_high).

モード:car(青), train(赤), bike(水色), walk(緑).

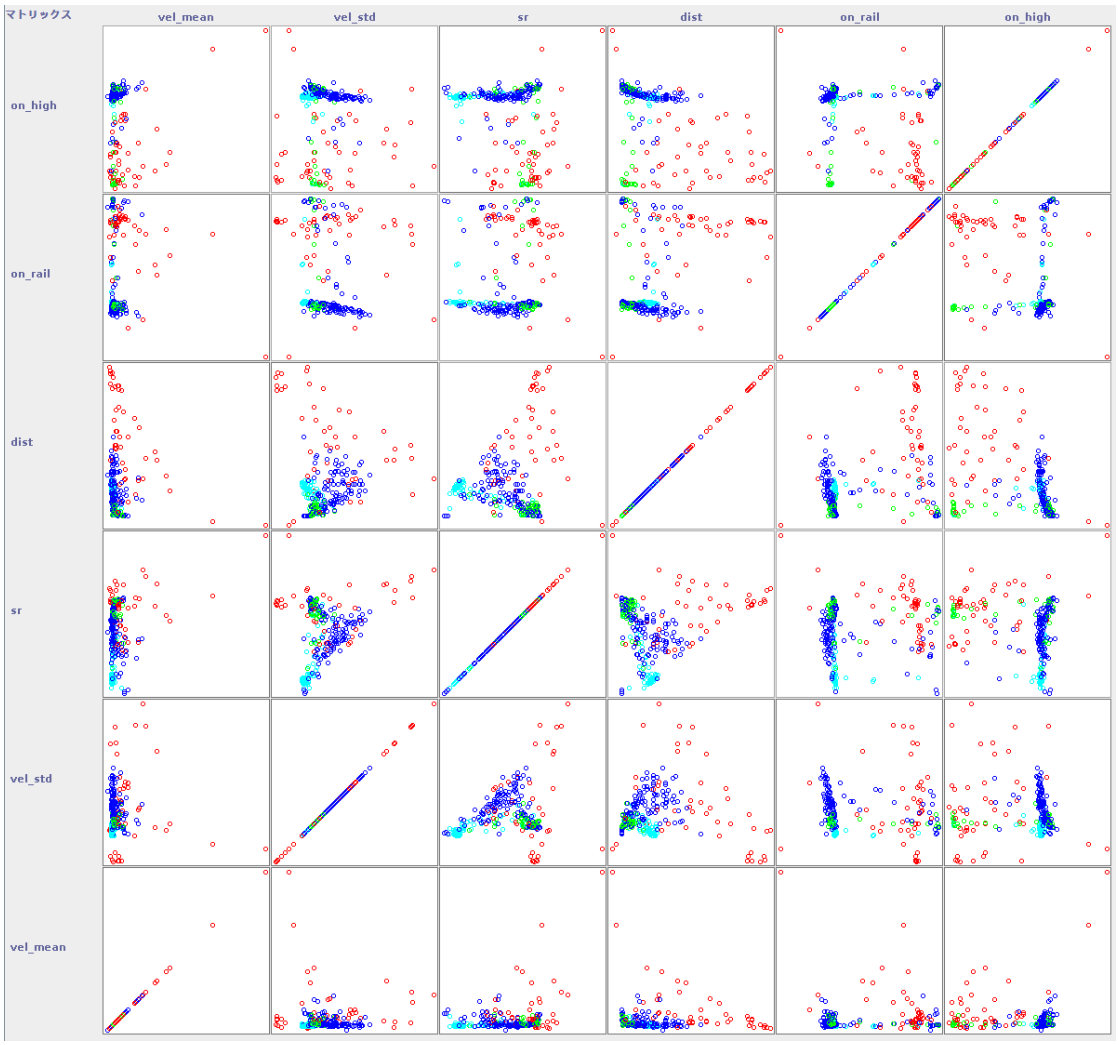

図 6.2: 不適切なラベル付きデータの分布.

特徴量:速度の平均 (vel_mean)・標準偏差 (vel_std), 停止点率 (sr), 距離 (dist), 線路とのマッチ率 (on_rail), 道路とのマッチ率 (on_high).

モード:car(青), train(赤), bike(水色), walk(緑).

6.6.2 誤推定に関して

本小節では, 交通移動モード推定システムの出力を実際に観察し, どのような点が推定誤りを起こしているのかを確認し, その修正方法に関して議論を行う.

表 6.27 は, ラベル付き軌跡の本数を 8 本とし, LP_GPS を用いて推定を行った実験の出力の詳細である. 表の行方向は推定されたモードを表していて, 列方向は真のモードを表している. 例えば, 1 行 1 列目は真のモードが walk であり推定結果も walk である点が 82774 点あることを表していて, 1 行 2 列目は真のモードが train であるが推定結果が walk である点が 307 点あることを表している. あるモード m に注目したとき, Recall とは,

$$Recall = \frac{\text{推定されたモードが } m \text{ であり,かつ真のモードも } m \text{ である点の個数}}{\text{真のモードが } m \text{ である点の個数}} \tag{6.2}$$

であり, Precision とは,

$$Precision = \frac{\text{推定されたモードが } m \text{ であり,かつ真のモードも } m \text{ である点の個数}}{\text{推定されたモードが } m \text{ である点の個数}} \tag{6.3}$$

である. 具体的には表 6.27 で, 対角成分の値を列方向に足しあわせた値で割ったものが recall であり, 対角成分の値を行方向に足しあわせた値で割ったものが precision である. 表 6.28 は, 表 6.27 をもとに Recall と Precision を計算した表である.

表 6.27: 出力結果の詳細 (点数).

Labeled\True	walk	train	bike	car
walk	82774	307	5597	14155
train	896	12889	139	863
bike	2525	48	37276	7726
car	16940	1065	14983	122592

表 6.27, 表 6.28 より 4 つのことが分かる.

第 1 に, train は他のモードと比較し precision と recall が共に高い点である. これは特徴量の線路とのマッチ率が, train とその他のモードを分離する良い特徴量であるからと考えられる.

第 2 に, 他のモードに対して bike の Recall が低いことである. この傾向は, 例で

表 6.28: 出力結果の詳細 (Precision/Recall).

—	walk	train	bike	car
Precision	0.80	0.87	0.78	0.79
Recall	0.80	0.90	0.64	0.84

取り上げた出力の場合だけでなく, 多くの出力において観察された傾向である. 4 章の図 4.1 からも予想できたように, bike を上手く分離する特徴量が存在しないことが, Recall を下げている原因の 1 つであると考えられる. 特に, bike と car の間の誤推定が主要な原因である. そのため, bike を上手く分離できる特徴量を発見することが, 推定精度向上への 1 つの手段となる.

第 3 に, 真のモードは walk であるが, bike や car と推定されている点が多いことである (表 6.27 の 1 列目). この原因は, ラベル付きデータの中に, 信号や渋滞などが原因で一時的に低速になっている bike や car のデータが存在するためであると考えられる. この問題を解決するには, car や bike にラベルを付ける際に, 速度が低速になっている箇所を削除すれば解決可能であると考えられる. ただし, そのような処理は, 第 4 の問題を大きくする原因となるの.

第 4 に, 推定されたモードは walk であるが, 真のモードは bike や car である点が多いことである (表 6.27 の 1 行目). この問題の原因は, 第 3 の問題と対になっており, ラベルなしデータの中に, 信号や渋滞などが原因で一時的に低速になっている bike や car のデータが存在するためであると考えられる. 解決方法として, 2.3.3 章で紹介した隠れマルコフモデルを後処理で使用する方法が考えられる. しかし, 実際に隠れマルコフモデルで後処理を行った結果 (表 6.29) を観察すると分かる通り, 後処理を行っても問題は依然として存在する. よって, この問題については今後の解決が必要となる.

6.7 まとめ

本章では, 実際の位置情報データを用いて提案システムと提案手法に関する評価実験を行った. 提案手法である LP_GPS は, ラベル付きデータの割合が全体の 10 % 以下の場合に, 既存手法に比べ平均して 2% から 3% の推定精度の向上を果たした. 一方で, LP_GPS は, ラベル付きデータの割合が少ない場合には RF や LP に比べ精度の標準偏差は大きい. これは, ラベルの付け方次第で推定精度に大きな差がでること

表 6.29: 隠れマルコフモデルで後処理を使用した場合の出力結果の詳細.

HMM 使用前				
Labeled\True	walk	train	bike	car
walk	94047	245	13762	23769
train	912	11734	45	430
bike	458	58	33506	4562
car	6458	2423	10579	115172
HMM 使用後				
Labeled\True	walk	train	bike	car
walk	94236	92	9651	16867
train	1583	14062	10	41
bike	588	0	40995	1607
car	5468	306	7236	125418

を示している. つまり, ラベル付きデータの割合が少ない場合は, 推定手法だけでなく, ラベルの付け方も重要である. 実験結果より, 特に walk/bike/car のモードに対して停止点率と距離の 2 つの特徴量の分布に注目することが, ゼロからラベル付作業を行う際の一つの指針となることが示された.

第 7 章

結論

7.1 本研究のまとめ

本研究では, ラベル付きデータの量が少ない場合でも, ラベル付きデータが十分にある場合と同等の推定精度で交通移動モード推定を行うことを目的とし, その目的を達成するために, 半教師あり学習手法を用いて交通移動モード推定を行った.

始めに, 既存の半教師あり学習手法である 4 手法 (Self-training model, Generative model, Co-training, Label propagation) を交通移動モード推定に適用し, 評価実験を行った. 実験の結果, 4 つの手法の中で Label propagation が交通移動モード推定に最も有効であることを確認した. 次に, Label propagation を交通移動モード推定用に拡張した推定手法 Label propagation for GPS を提案し, 提案手法を用いた交通移動モード推定システムを構築した.

提案手法の有効性を評価するため, 実際の位置情報データである GeoLife データセットを使用し, 推定実験を行った. 実験の結果, ラベル付きデータの割合が全体の 10% 以下と少ない場合に, 提案手法は既存手法に対して, 平均して 2% から 3% の推定性能の向上が確認され, また精度の標準偏差は大きいことが確認された. 標準偏差が大きいということは, ラベル付きデータの与え方次第で, 推定精度に大きな差がでるということであり, ラベル付きデータの割合が少ない場合は, どのデータに対しラベルを付けるかが重要となることがわかった. 実験結果から, 停止点率と距離の 2 つの特徴量における walk/bike/car の分布が明確に分離しているかを確認することが, ラベル付け作業の 1 つの指針になることが分かった.

7.2 今後の課題

まずはデータセットに関する課題である. 今回は GeoLife データセットのみを用いた実験であったが, 別のデータセットでの実験も行うべきである. その理由として, 第 1 に提案手法がどの程度のスケーラビリティの確認のためである. 今回の実験結果では特にラベルの割合が 10% 以下の場合に有効であることが示されたが, 軌跡の総数が 1,000 や 10,000 などに増加した場合でも同様のことが示せるかの確認は必要となる. 第 2 に地域ごとの交通手段の違いによる影響がどの程度あるのかの確認のためである. 地域ごとに交通移動モードの混合割合は異なるし, 各モードの平均速度などの特徴量も異なっているだろうと予想される. そのため, 有効となる特徴量が地域が変わっても変わらず有効であるかなどの確認が必要となる.

次に, 推定精度の向上に関する課題である. 1 つ目は, 特徴量に関してである. 本研

究で使用した特徴量では bike を十分に分離できないことが分かったので, bike を上手く分離可能な特徴量を探すことは重要である. 例えば, 位置情報以外に加速度計などのセンサーデータを用いる方向性がある. 2 つ目は, 推定に関してである. 本研究では, 特徴量不足のために十分な推定精度を出せなかった Co-training であるが, 車と徒歩と自転車を分離可能な地理的特徴量を見つけることが出来れば, 再考の余地がある. ただし, そのためには GPS の精度が現在より向上することが必要であると考えられる. 地理的特徴量ではなく, マイクロフォンを使用した音的特徴量などを使用しても良いかも知れない. LP_GPS を更に発展させる場合は, 重みの設定の仕方をより交通移動モード推定に適した形にする方向性がある. 例えば k 近傍グラフによる重み以外を考えることも意味があるかもしれない. また, 6.6.2 章で指摘した通り, walk と推定されたが, 真のモードは他である点が多数存在することで推定精度が落ちている問題の解決も必要である.

最後に, ラベルの付け方に関しする問題である. 実験では, ラベル付き軌跡の生成はランダムに行ったが, ラベルの割合が少ない場合は特に, どの軌跡にラベルを付けるかによって推定精度に大きな差が現れた. そのため少量のラベルで良い推定精度を出すためには, 推定手法を改良するだけでなく, ラベルの付け方も工夫する必要があると考えられる. ラベルの付け方に関しては 6.6 章の考察で触れたが, 今後, 別のデータセットを使用して 6.6 章の議論が別のデータセットでも当てはまるかの調査が必要となる.

謝辞

指導教員である安達淳教授には大変お世話になりました. 日々会議などで忙しい中, ミーティングでは毎週指導をしていただき, 時に厳しい時もありましたが先生のおかげでここまで頑張ることができました. ありがとうございます. サイバーフィジカルシステムプロジェクトを通じ, 国立情報学研究所の高須淳宏教授, 相原健郎准教授のお二人にも助言などを頂きお世話になりました. また秘書の久芳藍さん, 長尾美奈子さん, 開律子さんには研究以外でのサポートをして頂き, お世話になりました.

研究室の先輩であり博士課程の木村光樹先輩, 木下僚先輩には研究内容に関してだけではなく研究室や研究所の生活など様々なことでお世話になりました. 両先輩には, 輪講や論文の締め切り前には夜遅くまで残っていただき様々なアドバイスを頂きました, これらの事を乗り越えられたのもお二人の助けによるところが大きいです. また修士 1 年目には様々な勉強会で鍛えていただいたことにも感謝しています. 木下先輩には, 総合大会に提出する論文で苦しんでいる時に, 年末・年始にも関わらず研究所に来ていただき話を聞いてもらったことを大変感謝しております. ありがとうございました. 木村先輩も同様に, 提出期限ぎりぎりまで助力を頂き感謝しております. ありがとうございました.

自分を含め 2 人しかいませんでしたが, 同期の川勝孝也君には常に良い刺激を貰いました. 修士の間に国際会議という大舞台に挑戦し, 見事発表を成し遂げたことは素晴らしいと思います. 安達研究室最後の博士だと思いますが, 博士になっても頑張って下さい. また後輩である安東一慈君と西埜徹君は, 質問応答システムやグラフマイニングといった今の修士や博士の先輩がやっている研究とは少し違った事を頑張ろうとしていると思いますが, 先生方や博士の先輩方の力を借りながら, 無事に卒業をできる事を願っています. 頑張って下さい.

最後に研究以外のことで日々私を支えてくれた, 家族と友人達に感謝致します.

平成 28 年 2 月 4 日

芳賀 宣仁

参考文献

[AM06] Ian Anderson and Henk Muller. Practical activity recognition using gsm data. 2006.

[BCTH12] Adel Bolbol, Tao Cheng, Ioannis Tsapakis, and James Haworth. Inferring hybrid transportation modes from sparse gps data using a moving window svm classification. *Computers, Environment and Urban Systems*, Vol. 36, No. 6, pp. 526–537, 2012.

[BD+99] Kristin Bennett, Ayhan Demiriz, et al. Semi-supervised support vector machines. *Advances in Neural Information processing systems*, pp. 368–374, 1999.

[Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[BLvO13] Filip Biljecki, Hugo Ledoux, and Peter van Oosterom. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, Vol. 27, No. 2, pp. 385–407, 2013.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100. ACM, 1998.

[Bre01] Leo Breiman. Random forests. *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.

[BSS04] M Yu Byron, Krishna V Shenoy, and Maneesh Sahani. Derivation of kalman filtering and smoothing equations. 2004.

[BZM12] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 199–208. ACM,

2012.

[CCH+08] Tonmoy Choudhury, Sunny Consolvo, Brent Harrison, Jeffrey Hightower, Antonio Lamarca, Louis LeGrand, Azar Rahimi, Adam Rea, G Bordello, Bruce Hemingway, et al. The mobile sensing platform: An embedded activity recognition system. *Pervasive Computing, IEEE*, Vol. 7, No. 2, pp. 32–41, 2008.

[CCLS11] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. *ICWSM*, Vol. 2011, pp. 81–88, 2011.

[FDK+09] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A Landay. Ubigreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1043–1052. ACM, 2009.

[GBGC12] Martin Garbe, Christian Bunnig, Anne Gutschmidt, and Clemens Cap. Moving type detection without time information. In *Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on*, pp. 318–324. IEEE, 2012.

[GCBL11] Hongmian Gong, Cynthia Chen, Evan Bialostozky, and Catherine T Lawson. A gps/gis method for travel mode detection in new york city. *Computers, Environment and Urban Systems*, Vol. 36, No. 2, pp. 131–139, 2011.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, Vol. 3, pp. 1157–1182, 2003.

[geo] "`http://research.microsoft.com/en-us/projects/GeoLife/`".

[gps] Gps accuracy. "`http://www.gps.gov/systems/gps/performance/accuracy/`".

[GWB+09] Paola A Gonzalez, Jeremy S Weinstein, Sean J Barbeau, M Labrador, Philip L Winters, Nevine L Georggi, and R Perez. Automating mode detection for travel behaviour analysis by using global positioning systemsenabled mobile phones and neural networks. *Intelligent Transport Systems, IET*, Vol. 4, No. 1, pp. 37–49,

2009.

[HTFF05] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, Vol. 27, No. 2, pp. 83–85, 2005.

[Joa99] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, Vol. 99, pp. 200–209, 1999.

[KIIF10] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. Travel route recommendation using geotags in photo sharing sites. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 579–588. ACM, 2010.

[LWY+10] Xin Lu, Changhu Wang, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the international conference on Multimedia*, pp. 143–152. ACM, 2010.

[MCJ06] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pp. 152–159. Association for Computational Linguistics, 2006.

[MEBH08] M Mun, Deborah Estrin, Jeff Burke, and Mark Hansen. Parsimonious mobility classification using gsm and wifi traces. In *Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets)*, 2008.

[MRS+09] Min Mun, Sasank Reddy, Katie Shilton, Nathan Yau, Jeff Burke, Deborah Estrin, Mark Hansen, Eric Howard, Ruth West, and Peter Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proceedings of the 7th international conference on Mobile systems, applications, and services*, pp. 55–68. ACM, 2009.

[PSR+13] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, et al. Se-

mantic trajectories modeling and analysis. *ACM Computing Surveys (CSUR)*, Vol. 45, No. 4, p. 42, 2013.

[RBE+08] Sasank Reddy, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Determining transportation mode on mobile phones. In *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pp. 25–28. IEEE, 2008.

[RHS05] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005.

[RMB+10] Sasank Reddy, Min Mun, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava. Using mobile phones to determine transportation modes. *ACM Transactions on Sensor Networks (TOSN)*, Vol. 6, No. 2, p. 13, 2010.

[RV95] Joel Ratsaby and Santosh S Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the eighth annual conference on Computational learning theory*, pp. 412–417. ACM, 1995.

[See00] Matthias Seeger. Learning with labeled and unlabeled data. Technical report, 2000.

[STSA12] Leon Stenneth, Kenville Thompson, Waldin Stone, and Jalal Alowibdi. Automated transportation transfer detection using gps enabled smartphones. In *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, pp. 802–807. IEEE, 2012.

[SVL+06] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G Griswold, and Eyal De Lara. Mobility detection using everyday gsm traces. In *UbiComp 2006: Ubiquitous Computing*, pp. 212–224. Springer, 2006.

[SWYX11] Leon Stenneth, Ouri Wolfson, Philip S Yu, and Bo Xu. Transportation mode detection using mobile phones and gis information. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 54–63. ACM, 2011.

[WCM10] Shuangquan Wang, Canfeng Chen, and Jian Ma. Accelerometer based transportation mode recognition on mobile phones. In *Wearable Computing Systems (APWCS), 2010 Asia-Pacific Conference on*,

pp. 44–46. IEEE, 2010.

[WHO+13] Apichon Witayangkurn, Teerayut Horanont, Natsumi Ono, Yoshihide Sekimoto, and Ryosuke Shibasaki. Trip reconstruction and transportation mode extraction on low data rate gps data from mobile phone. In *Proceedings of the international conference on computers in urban planning and urban management (CUPUM 2013)*, pp. 1–19, 2013.

[WNB12] Peter Widhalm, Philippe Nitsche, and N Brandie. Transport mode detection with realistic smartphone sensor data. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 573–576. IEEE, 2012.

[XSG+11] Dafeng Xu, Guojie Song, Peng Gao, Rongzeng Cao, Xinwei Nie, and Kunqing Xie. Transportation modes identification from mobile phone data using probabilistic models. In *Advanced Data Mining and Applications*, pp. 359–371. Springer, 2011.

[YYLL11] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pp. 325–334. ACM, 2011.

[YZXW10] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. Smart itinerary recommendation based on user-generated gps trajectories. In *Ubiquitous Intelligence and Computing*, pp. 19–34. Springer, 2010.

[ZCL+10] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. Understanding transportation modes based on gps data for web applications. *ACM Transactions on the Web (TWEB)*, Vol. 4, No. 1, p. 1, 2010.

[ZCXM09] Yu Zheng, Yukun Chen, Xing Xie, and Wei-Ying Ma. Geolife2. 0: a location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pp. 357–358. IEEE, 2009.

[ZG02] Zhu and Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107,

2002.

[ZG09] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, Vol. 3, No. 1, pp. 1–130, 2009.

[ZGL+03] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, Vol. 3, pp. 912–919, 2003.

[ZLC+08] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. Understanding mobility based on gps data. In *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 312–321. ACM, 2008.

[ZP13] Zelun Zhang and Stefan Poslad. A new post correction algorithm (pocoa) for improved transportation mode recognition. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pp. 1512–1518. IEEE, 2013.

[ZZ11] Yu Zheng and Xiaofang Zhou. Computing with spatial trajectories. 2011.

[ZZXM09] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pp. 791–800. ACM, 2009.

[ZZXY10] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pp. 1029–1038. ACM, 2010.

[足立 12] 足立修一, 丸田一郎. カルマンフィルタの基礎. 東京電機大学出版局, 2012.

[日野 14] 日野高志. 位置情報の活用で進化するケータイサービス. "`http://www.kddi-ri.jp/article/RA2014001`", 2014.

[飛田 02] 飛田幹男. 世界測地系と座標変換. 社団法人日本測量協会, 2002.

発表文献

国内会議

[1] 芳賀宣仁, 木下僚, 安達淳.「軌跡データを用いた半教師有り学習による交通手段推定」2016 年 電子情報通信学会 総合大会 (IEICE2016), 福岡, 2016 年 3 月．(発表予定)

付録 A

データセット

表 A.1: GeoLife データ概要

全体	ユーザー数	ポイント数	plt ファイル数
	182 users	24,876,978	18,670
Label 有り	ユーザー数	ポイント数	plt ファイル数
	64 users[*2]	5,433,669	4477
Label 有りデータの割合	34.1%	21.8%	24.0%

表 A.2: モードデータ概要

モード名	walk	subway	car	bike	motorcycle
ポイント数	1,587,421	286,455	513,490	949,672	338
データの割合	29.215%	5.272%	9.450%	17.478%	0.006%
airplane	taxi	train	boat	run	bus
9,183	242,453	560,979	3,559	1971	1,278,148
0.169%	4.462%	10.324%	0.065%	0.036%	23.523%

本章では実験データの詳細に関して述べる.

A.1 GeoLife データセット

本研究の実験で使用したデータセットは Microsoft Research の GeoLife GPS Trajectories [ZLC+08, ZZXM09, ZCXM09] [*1] (以下 GeoLife) である.

表.A.1 は GeoLife データ全体の情報を示している. plt ファイルにはユーザー 1 人の 1 日分のデータが含まれている. またモードは全部で 11 種類存在し, その割合は表.A.2 の通りである. GPS ポイントのデータ形式は表.A.3 の通りとなる. なお GeoLife のラベル付きデータは点に直接ラベルが付いているのではなく, ある軌跡のある時間からある時間までが何のモードであったかを示すファイルが別途用意されているので, ラベルはそのファイルに従い付与した.

[*1] http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/

表 A.3: データ形式

列番号	項目
0	緯度 (double/10 進数/小数点以下 6 桁が最大)
1	経度 (double/10 進数/小数点以下 6 桁が最大)
2	未使用 (常に 0)
3	標高 (int/整数値/-777 はエラー値)
4	1899-12-30 からの経過日数 (double/小数点以下 10 桁が最大)
5	日にち (String)
6	時間 (String)

A.2 実験データ生成

本稿で行った実験では GeoLife のデータを以下の方針で作成・整形したデータを用いた.

1. 点と点の時間間隔が 180 秒以上ある場合に別の軌跡と考え軌跡を生成する.
2. 100 点以上の点を含み, かつ全ての点にラベルが付与されている軌跡を抽出[*2].
3. 中国北京市内の軌跡に限定 (緯度 39.837521～40.109353, 経度 116.2373198～116.5324913 の東西 約 25km × 南北 約 30km の範囲).
4. モードを 4 種類にまとめる. walk(=walk+run), car(=car+taxi+bus), bike(=bike), train(=train+subway), 他は未使用
5. エラー軌跡の除去.(Walk ラベルであるのに 15[m/s] の軌跡や, Train ラベルであるのに明らかに線路上を走っていない軌跡など)

以上の処理を行い生成したデータセットが表 A.4 である. また図 A.1 はデータセットの GPS ポイントを地図上にプロットした図である.

[*2] 軌跡の一部のみラベルが付いているという場合が多数存在した.

表 A.4: 実験データセット

—	walk	train	bike	car	total
ポイント数	103,849	14,809	58,983	147,235	324,876
データの割合	31.96%	4.56%	18.15%	45.32%	100.00%
軌跡数	588	–	–	–	–

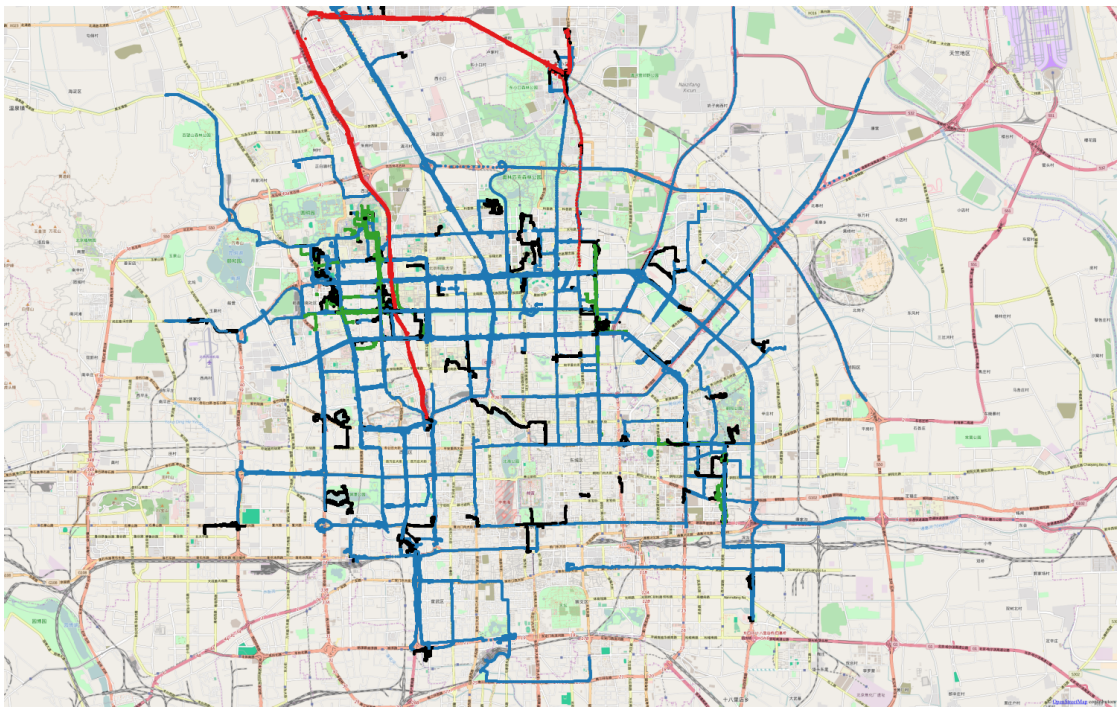

図 A.1: 実験データセットの地図上の分布.

青点:car, 赤点:train, 黒点:walk, 緑点:bike. (Map data ©OpenStreetMap contributors, CC BY-SA)

付録 B

カルマン・フィルター

カルマン・フィルター [BSS04, 足立 12] は時系列データに対する線型推定法の一種である. ある現象の観測値データを $y_1, y_2, ..., y_t$ とする. この各 y_t は直接観測することが出来ない変数 x_t に依存している. x_t は状態変数と呼ばれ, 観測可能な y_t から観測不可能な状態 x_t を推定することがカルマン・フィルターの目的である.

カルマン・フィルターのモデルは次の通りである.

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{w}_t \tag{B.1}$$
$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t \tag{B.2}$$

ここで $\mathbf{x}_t$ は n 次元ベクトルであり, $\mathbf{y}_t$ は p 次元ベクトルである. 更に $\mathbf{A}, \mathbf{C}$ は $n \times n, p \times n$ 行列である. 式.B.1 は状態方程式またはシステム方程式と呼ばれ, この式に従い状態 $\mathbf{x}_t$ は変化する. なお $\mathbf{w}_t$ は状態方程式の誤差であり, 平均値ベクトル $\mathbf{0}$, 共分散行列 $\mathbf{Q}$ の正規白色雑音である. 式.B.2 は観測方程式と呼ばれ, 状態が $\mathbf{x}_t$ の時に観測値 $\mathbf{y}_t$ は, この方程式に従い観測される. なお $\mathbf{v}_t$ は平均値ベクトル $\mathbf{0}$, 共分散行列 $\mathbf{R}$ の正規白色雑音である.

ここで事前推定値と事後推定値を定義する.

- 事前推定値 : $\hat{\mathbf{x}}^-(t)(= \hat{\mathbf{x}}^-(t|t-1))$
  時刻 $t-1$ までのデータを用いた, 時刻 t での $\mathbf{x}$ の推定値 (予測推定値).
- 事後推定値 : $\hat{\mathbf{x}}(t)(= \hat{\mathbf{x}}(t|t))$
  時刻 t までのデータを用いた, 時刻 t での $\mathbf{x}$ の推定値 (フィルタリング推定値). この値が目的となる推定値である.

カルマン・フィルタでは次の 2 ステップを交互に実行することで, 各時点の状態推定値 $\hat{\mathbf{x}}(t)$ と誤差共分散行列を得る.

- 予測ステップ (predict)

$$\text{事前状態推定値} : \hat{\mathbf{x}}^-(t) = \mathbf{A}\hat{\mathbf{x}}(t-1) \tag{B.3}$$
$$\text{事前誤差行分散行列} : \mathbf{P}^-(t) = \mathbf{A}\mathbf{P}(t-1)\mathbf{A}^T + \mathbf{Q} \tag{B.4}$$

- フィルタリングステップ (filtering)

$$\text{カルマンゲイン行列} : \mathbf{G}(t) = \mathbf{P}^-(t)\mathbf{C}^T(\mathbf{C}\mathbf{P}^-(t)\mathbf{C}^T + \mathbf{R})^{-1} \tag{B.5}$$
$$\text{状態推定値} : \hat{\mathbf{x}}(t) = \hat{\mathbf{x}}^-(t) + \mathbf{G}(t)(\mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}^-(t)) \tag{B.6}$$
$$\text{事後誤差共分散行列} : \mathbf{P}(t) = (\mathbf{I} - \mathbf{G}(t)(C))\mathbf{P}^-(t) \tag{B.7}$$

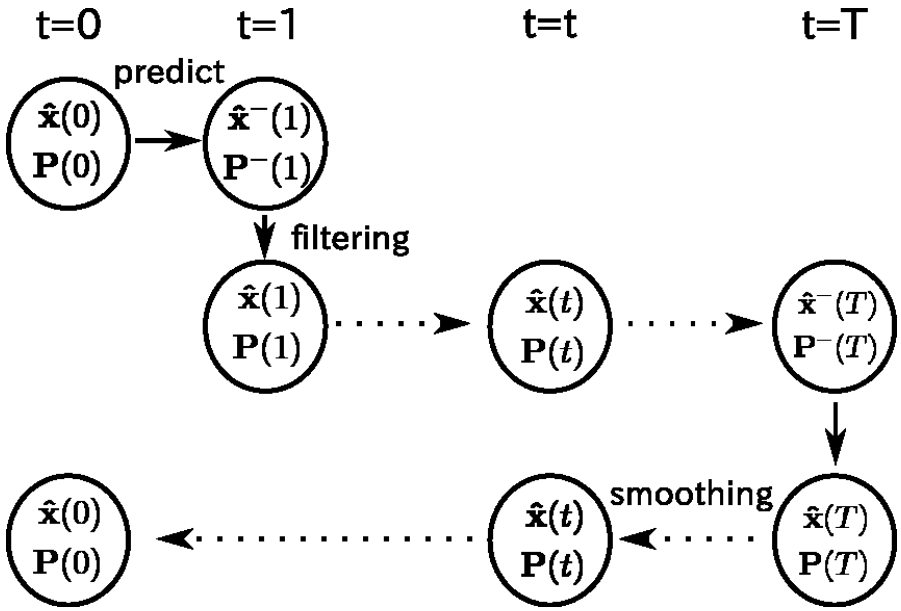

図 B.1: Kalman Filter and Smoother

予測ステップでは一時点前 $(t-1)$ の予測値と状態方程式から, 現時点 (t) の事前推定値を計算する. 更にフィルタリングステップでは新たな観測 $\mathbf{y}(t)$ を利用して, 求めたい状態推定値を計算する. なお初期値に関しては $\hat{\mathbf{x}}(0) \sim N(\mathbf{x}_0, \sum_0)$ として, $\sum_0 = \mathbf{P}_0 = \gamma\mathbf{I}$($\gamma$ は調整パラメータ) などとすることが多い.

次に, カルマン・スムーザーについて説明する. カルマン・フィルターでは現時点 t までに得られた観測値 $\mathbf{y}_1, ..., \mathbf{y}_t$ を用いて状態 $\mathbf{x}_t$ を推定しが, カルマン・スムーザーでは観測値系列$\{\mathbf{y}_t\}(t = 1, ..., T)$ が全て得られた後に, 時点をさかのぼり状態 $\mathbf{x}_t$ を推定する. フィルタリングとスムージングの違いは, 図.B.1 のように予測・フィルタリングは右方向と下方向の矢印で表される処理を行うのに対し, スムージングは左方向の矢印で表される処理を行う.

スムージングでは, 以下の 3 式を用いてフィルタリングとは逆に時刻の巻き戻る方向へ処理を行い, 推定値を得る.

$$\mathbf{J}(t) = \mathbf{P}(t)\mathbf{A}^T(\mathbf{P}^-(t+1))^{-1} \tag{B.8}$$

$$\mathbf{x}(t|T) = \mathbf{x}(t) + \mathbf{J}(\mathbf{x}(t+1|T) - \mathbf{A}\mathbf{x}(t)) \tag{B.9}$$

$$\mathbf{P}(t|T) = \mathbf{P}(t) + \mathbf{J}(t)(\mathbf{P}(t+1|T) - \mathbf{P}(t+1))\mathbf{J}(t)^T \tag{B.10}$$

$\mathbf{P}(t)$ は現時点 t までの観測値 $(t = 1, ..., t)$ が得られたうえでの時刻 t の推定値であり, $\mathbf{P}(t|T)$ は全ての観測値 $(t = 1, ..., T)$ が得られたうえでの時刻 t の推定値を表す. この $\mathbf{P}(t|T)$ がスムージングされた推定値となる.

付録 C

混合ガウスモデルと EM アルゴリズム

C.0.1 混合ガウスモデル

混合ガウスモデル (Gaussian Mixture Model, GMM)[Bis06] は、次のようなガウス分布の重ね合わせで表されるモデルである.

$$p(\mathbf{x}) = \sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sum_k). \tag{C.1}$$

ここで K は混合数と呼ばれ重ね合わせるガウス分布の個数を示す. π_k は混合係数と呼ばれ π_k は $0 \leqq \pi_k \leqq 1, \sum_{k=1}^{K} \pi_k = 1$ を満たす.

ここでは今後の説明の為, 離散的な潜在変数を用て上記の混合ガウスモデルについて説明する. まず K 次元の 2 値確率変数 $\mathbf{z}$ を導入する. $\mathbf{z}$ は どれか一つの要素 z_k のみが 1 で他は 0 を取るベクトルである. つまり $\mathbf{z}$ は K 個の離散的な状態を取る. ここで周辺分布 $p(\mathbf{z})$ と条件付き分布 $p(\mathbf{x}|\mathbf{z})$ は,

$$p(z_k = 1) = \pi_k \tag{C.2}$$

$$p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sum_k). \tag{C.3}$$

であるので, 同時分布 $p(\mathbf{x}, \mathbf{z})$ はこれらの積で表すことが出来る. ゆえに $\mathbf{x}$ の周辺分布は, $\mathbf{z}$ の全ての状態に関して足しあわせて,

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sum_k). \tag{C.4}$$

となる. 潜在変数の意味としては, 複数の観測データ $\mathbf{x}_1, ..., \mathbf{x}_N$ が与えられた場合, 各観測データ $\mathbf{x}_n$ について, 対応する分布がどの分布かを示す潜在変数 $\mathbf{z}_n$ が存在するということである.

また次の節で重要になる量として, $\mathbf{x}$ が与えられた下での条件付き確率 $p(z_k|\mathbf{x})$ を定義する.

$$\gamma(z_k) \equiv p(z_k = 1|\mathbf{x}) = \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \sum_j)} \tag{C.5}$$

π_k は $z_k = 1$ となる事象の事前確率, $\gamma(z_k)$ は $\mathbf{x}$ を観測した後での事後確率とみなせる. この $\gamma(z_k)$ は負担率 (responsibility) と呼ばれ, 混合要素 k が $\mathbf{x}$ を観測する度合いを表していると解釈できる.

C.0.2 EM アルゴリズム

EM アルゴリズム (expectation-maximization algorithm) とは, 潜在変数を持つモデルの最尤解を求める方法の一つである. ここでは特に GMM のパラメータ推定にしぼった EM アルゴリズムについて説明する.

観測したデータ集合 $\mathbf{x}_1, ...\mathbf{x}_N$ に対応する混合ガウスモデルのパラメータを推定する問題を考える. 各データ点が混合ガウス分布から独立に生成されたとすると, 対数尤度関数は,

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \sum) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \sum)\}. \tag{C.6}$$

となる. 解きたい問題は, この対数尤度関数を最大とするパラメータを得ることである.

尤度最大化において満たされる条件は, 対数尤度の式.C.6 を平均 $\boldsymbol{\mu}_k$, 分散 $\sum_k$, 混合率 π_k に関して微分した値が 0 となることである. ただし, 混合率 π_k に関しては $\sum_{k=1}^{K} = 1$ という制約条件があるため, ラグランジュの未定乗数法を使用する. 微分して整理した結果は次の通りとなる.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(x_{nk}) \mathbf{x}_n \tag{C.7}$$

$$\sum_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \tag{C.8}$$

$$\pi_k = \frac{N_k}{N} \tag{C.9}$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{C.10}$$

これらの結果より, モデルの平均と分散は負担率による重み付き平均と分散になることが理解できる. 混合係数は, その要素の全データ点に対する負担率の平均である.

しかし負担率の式を見ると分かるように, これらのパラメータは陽な解を与えてはいない. そこで EM アルゴリズムでは E ステップと M ステップという 2 つの更新手続きを繰り返し行うことでパラメータを推定する.

1. 平均 $\boldsymbol{\mu}_k$, 分散 $\sum_k$, 混合係数 π_k を初期化し対数尤度の初期値を計算する.
2. **E-Step**:

現在のパラメータ値を使用し, 負担率を計算する.

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \sum_j)}$$

3. **M-Step**:
   現在の負担率を使用し, パラメータ値を再計算する.

$$\begin{aligned}
\boldsymbol{\mu}_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(x_{nk}) \mathbf{x}_n \\
\sum_k^{new} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})(\mathbf{x}_n - \boldsymbol{\mu}_k^{new})^T \\
\pi_k^{new} &= \frac{N_k}{N} \\
N_k &= \sum_{n=1}^{N} \gamma(z_{nk})
\end{aligned}$$

4. 対数尤度 $\ln p(\mathbf{X}|\boldsymbol{\mu}, \sum, \text{ß}$ を計算し, パラメータ値の変化, もしくは対数尤度の変化を見て収束性を確認し, 条件を満たしていない場合は E-step へ戻る.

C.0.3 EM アルゴリズム for Semi-supervised Gaussian Mixture Model

Semi-supervised Gaussian Mixture Model(SGMM) とはラベル付きデータとラベルなしデータの両方を用いて, 混合ガウス分布のパラメータを推定する半教師あり学習手法である.

ラベル付きデータを $L = (\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)$, ラベルなしデータを $U = \mathbf{x}_{l+1}, ..., \mathbf{x}_{l+u}$ としてデータ全体を $D = L + U$ とする. SGMM ではラベルありデータとラベルなしデータの両方を使用して, 対数尤度 $\log(D|\theta) = \sum_{i=1}^{l} \log p(y_i|\theta) p(\mathbf{x}_i|y_i, \theta) + \sum_{i=l+1}^{l+u} \log p(\mathbf{x}_i|\theta)$ の最大化を考える.

パラメータ推定は EM アルゴリズムで行われ, 教師なし学習の場合 (前小節) との違いは負担率の計算の部分である. ラベルなしデータに関しては前小節と同様に負担率は $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \sum_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \sum_j)}$ で計算する, 一方でラベルありデータに関しては $\gamma(z_{nk}) = 1$(k がクラス n に属する), $\gamma(z_{nk}) = 0$(それ以外) となる.