Vol.2013-DBS-158 No.24
2013/11/26

整数計画法を用いた GPSログからの滞留点と訪問POIの同時推定

数原 良彦[1,a)]　戸田 浩之[1]　西川 仁[2]　鷲崎 誠司[1]

概要：GPS などにより得られる位置情報を利用して，訪問 POI 推定の研究が行われてきた．しかしながら，従来技術においては，GPS ログに対して滞留点抽出の処理を行い，その結果得られた滞留点に対して訪問 POI 推定を行う直列処理を利用している．このため，滞留点抽出誤りがその後の訪問 POI 推定の精度に直接影響する問題があった．本稿においては，滞留点抽出，訪問 POI 推定，訪問 POI の遷移，訪問 POI 数の妥当性を目的関数に取り入れて整数計画問題として定式化することで，滞留点抽出と訪問 POI 推定を同時に実現する枠組みを提案する．

1. はじめに

スマートフォンの普及に伴い，位置情報を用いたサービスが広く使われるようになっている．これらの位置情報サービスではスマートフォンが GPS を用いて測位した位置情報を利用している．このように個人が持つ端末を用いた GPS 測位とその位置情報の利用に対する敷居が下がってきており，最近では Google Now[*1] のようにユーザの位置情報を端末から送信することで，周辺情報や交通情報などの情報を受け取ることができるサービスが普及しつつある．その他に，しゃべってコンシェル[*2] や Siri[*3]のようにスマートフォン向けに設計された対話エージェントを通じてユーザ状況に合わせたパーソナルアシスタントが存在する．

このような位置情報を用いたサービスを高度化し，ユーザ状況に合わせて情報提供するためには，ユーザが現在どのような状態であるのか推定するユーザコンテキスト推定が重要な課題となる．我々は，GPS 測位または WiFi 測位によって得られる移動履歴のような信号レベルの情報に対して，ユーザ状況や興味などの意味レベルのコンテキスト情報を自動的に付与することを目指している．たとえば，ユーザが移動中であるか，仕事中であるかというユーザ状況推定や，ユーザが訪れた Point of Interest (POI) を自動的に判定することで，ユーザの嗜好分析が可能となる．最初の段階として，ユーザの訪問 POI 推定という課題に取り組む．たとえば，ユーザの移動履歴に対して，当該ユーザが訪問した を自動的に推定することが可能になれば，ユーザの嗜好分析や定期的に訪問する POI 訪問を忘れた場合にパーソナルアシスタントが高精度にリマインダを行うことが可能となる．

ユーザの訪問 POI 推定にはいくつか既存の取り組みがある [14][12][9]．西田らの方法 [14] は GPS ロガーやネットワーク位置情報源を利用して得られたユーザの位置履歴から滞留点抽出を行い，抽出された滞留点に対して訪問 POI を推定する二段階の処理で訪問 POI 推定を実現している．

しかしながら　従来手法においては訪問 POI 推定が滞留点抽出の結果をそのまま利用する直列処理であるため，滞留点抽出の誤りがそのまま訪問 POI 推定の誤りとして伝播するという問題がある．訪問 POI が存在しない地点において滞留点が誤って抽出された場合に，どのような訪問 POI を推定しても誤りとなり，また逆に，滞留点抽出漏れの場合においては，そもそも訪問 POI 推定ができないため，必ず誤りとなる (図 1 の右参照)．

我々は滞留点抽出と訪問 POI 推定を同時に扱う枠組みを導入することにより，上記の 2 つの課題を解決し，訪問 POI 推定精度の更なる向上が可能になると考えた．本稿では，滞留点抽出と訪問 POI 推定を 0-1 整数計画問題として定式化し，汎用ソルバを用いて求解する方法を用いることで，遷移情報を考慮しながら滞留点抽出と訪問 POI 推定を同時に解く方法を提案する．既に多くの応用分野において整数計画法を用いた手法が提案されており，汎用ソルバの

[1] 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
[2] 日本電信電話株式会社 NTT メディアインテリジェンス研究所
NTT Media Intelligence Laboratories, NTT Corporation
[a)] suhara.yoshihiko@lab.ntt.co.jp
[*1] http://www.google.com/landing/now/
[*2] http://www.nttdocomo.co.jp/service/information/shabette_concier/
[*3] http://www.apple.com/ios/siri/

性能向上により，実用的な時間で求解が可能であることが示されている．Roth ら [8] は自然言語処理における固有表現抽出と固有表現間の関係抽出を整数計画問題として表現し，これを同時に解くことで精度が向上することを示した．Roth らが扱った課題は本稿における滞留点抽出と訪問 POI 推定の同時推定という課題と基本的には同じ構造であり，本稿で扱う課題も同時推論の対象とすることは自然であると考えた．提案手法においては，滞留点の確からしさを表すスコア，訪問 POI 候補のもっともらしさを表すスコア，訪問 POI の遷移のもっともらしさの観点を整数計画問題の目的関数に取り入れることによって，滞留点と訪問 POI 系列としてもっともらしい変数値の組み合わせが解として得られるような定式化を行う．

本稿では，GPS などによって得られる移動履歴からユーザコンテキスト推定を実現するため，以下の 2 つの Research Question (RQ) の解決に取り組む．

RQ1. 高精度な訪問 POI 推定を実現する最適な滞留点抽出手法は何か?

RQ2. 訪問 POI 推定と滞留点抽出を同時に行うことは可能か?

RQ1 については，既存研究で様々な滞留点抽出手法 [2][1][12][14][5] が提案されているが，いずれも問題に合わせたパラメータ設定によってアルゴリズムの調整が必要であり，単一のパラメータで高精度な訪問 POI 推定を実現する滞留点抽出手法は存在しないと考えられる．また，既存研究においては滞留点抽出の結果に対して訪問 POI 推定を行う方法が提案されており，RQ2 は未着手の課題である．

本稿の貢献は以下のとおりである:

- 既存の滞留点抽出手法において，1 つの固定されたパラメータでは訪問 POI が存在する滞留点抽出を高い適合率で網羅的に抽出することが困難であることを明らかにする．
- 滞留点抽出と訪問 POI 推定を整数計画問題として定式化することにより，遷移情報を考慮しながら同時推定を実現する手法を提案する．

本稿の構成は以下のとおりである．2 章で関連研究について述べ，3 章で本稿で扱う語の定義と問題設定を説明する．4 章で評価実験に用いるデータセットに対する予備分析によって得られた知見を述べる．5 章で提案手法について述べる．6 章で評価実験を述べ，7 章で本稿をまとめる．

2. 関連研究

ユーザが蓄積した GPS ログなどを用いたユーザ状態の推定の研究としては，滞留点抽出 [2][1][14]，移動モード判定 [10][13]，訪問 POI 推定 [6][14][9] などが挙げられる．

滞留点抽出としては位置履歴集合に対してクラスタリング手法を適用することによって抽出する方法が一般的である．Ashbrook [2] は K-Means を，Adams ら [1] は DBSCAN [4] を利用している．西田ら [14] は Mean-Shift [3] を時間軸に拡張した時空間 Mean-Shift を利用している．Kang ら [5] や Zheng ら [12] は測位点集合の重心から一定の半径以内に一定時間以上測位点集合が存在する場合に滞留点抽出を行う手法を利用している．いずれの手法においても滞留点抽出のために閾値等のパラメータ設定が必要であり，滞留点抽出の適合率を重視すれば再現率を損ねるおそれがある．

ユーザの履歴情報から訪問 POI を推定する取り組みもある [6][14][9]．Lian ら [6] や Shaw ら [9] は，ユーザの現在位置に対して付近の POI をランキングして提示する手法を提案している．彼らはチェックインサービスにおけるユーザのチェックイン履歴を教師データとして利用して教師ありランキング学習 [7] を行い，ランキングの最適化をしている．西田らは，ユーザは POI カテゴリを選択した後に訪問 POI を決定するという仮定と訪問 POI カテゴリによって滞在時間が異なるという仮定に基づいた訪問 POI の生成モデルを提案している．この方法では，一部の滞留点に対してラベルが付与された場合においては，半教師あり学習を実現することが可能である．これら既存の訪問 POI 推定に関わる研究では滞留点抽出は前処理で行われるものとしており，滞留点抽出漏れや誤検出を訪問 POI 推定と同時に扱う研究は我々の知る限り存在せず，本稿の新規性はこの点にある．

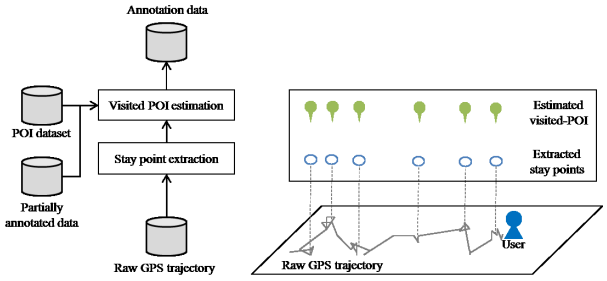

図 1 問題設定の概略図

表 1 データセットに含まれる訪問 POI カテゴリ毎の統計値 (*印はカスタム POI)

Category	#SP	#POI	mean (sp)	sd (sp)
Home (*)	30	2	19683.4	12773.9
Train station	20	4	378.2	178.6
Bus station	11	2	289.6	96.8
Office	11	1	35023.2	5063.7
All	98	24	10522.1	14208.3

3. 問題設定

本稿では，ユーザの移動履歴に加えて，一部の移動履歴に対して滞留点や訪問 POI のアノテーションが付与されている状況を想定する．これらの情報と POI データベースが与えられ際に，システムはユーザの移動履歴を元にい

つ，どのような POI を訪問したのかという推定を行う．図 1 に問題設定の概略図を示す．

最近では，スマートフォンやタブレットの普及とこれらの端末のほとんどが GPS 測位，WiFi 測位機能を有することから，ユーザが普段から持ち歩いている端末によって，ユーザの移動履歴が利用可能な形で日々蓄積している状況を考える．また，訪問 POI のアノテーションには，たとえば Foursquare[*4]やロケタッチ[*5]のような既存の位置情報サービスの機能を利用することもできる．これらのサービスでは，ユーザが訪問した POI をリアルタイムに発信する機能が実装されており，これらのサービス単体で，あるいは Twitter[*6]などの他のサービスと連携した形で利用されている．

移動履歴に関わる情報をユーザを限定せずに公開すると，見知らぬ他人が個人の移動履歴からユーザの個人情報に関わる情報を推測することが可能であり，プライバシーの問題が発生するおそれがある．そこで，本稿ではユーザが付与したアノテーションデータのみを用いて，当該ユーザの高度なコンテキスト推定実現を目指す．

定義 1 (測位点). GPS やネットワーク位置情報源から得られる測位点 g_i は経度，緯度，タイムスタンプ $g_i = (lng_i, lat_i, t_i)$ から構成される．GPS 以外には WiFi による測位などが挙げられ，広く普及しているスマートフォン，タブレットでは双方を用いた位置測位が一般的に利用されている．本稿では測位点の取得方法については特に区別しないものとする．

ユーザが端末の測位機能を有効にした状態で日々の生活を過ごすことで，連続する測位点集合の履歴を記録することができる．たとえば 1 日単位というように，測位点集合を意味のある区間に分けることが可能である．

定義 2 (セッション). セッション $\mathbf{g}_j = \{g_i\}_{i=n_{j-1}+1}^{n_j}$ はユーザが持つ測位点集合 $\mathbf{G} = \{g_i\}_{i=1}^{N}$ のうち，連続する測位点の系列単位である．各セッションは連続する排他的な集合であるため $\mathbf{G} = \coprod_n \mathbf{g}_n$ である．本稿では，セッション単位を 1 日とし，同じ日付の 0:00 から 23:59 の時刻に記録された測位点集合を 1 セッションとして用いる．例えば，あるユーザの端末に記録されている全ての測位点の集合が $\mathbf{G}$ であり，当該ユーザのある 1 日の測位点集合 $\mathbf{g}_j$ として表される．

定義 3 (滞留点候補). 一定時間同じ位置に滞在したことを表す測位点集合．滞留点候補 sp_n は複数の連続する測位点から構成され，それらの重心によって中心座標 $(\mathrm{sp}_n.lng, \mathrm{sp}_n.lat)$ を表す．滞留点候補に含まれる測位点から開始時刻 $\mathrm{sp}_n.bt$，終了時刻 $\mathrm{sp}_n.et$ および滞留時間 $\mathrm{sp}_n.st$ を求めることが可能である．

*4 http://foursquare.com/
*5 http://tou.ch/
*6 http://twitter.com/

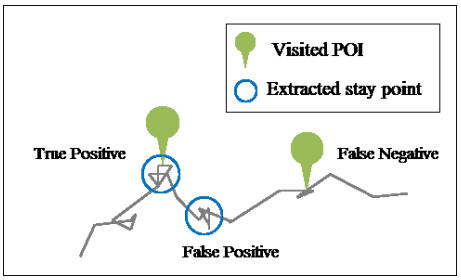

図 2 移動履歴からの滞留点抽出の正解と誤り

定義 4 (意図した滞留点). 意図した滞留点 sp_n は，ユーザが自分の意思でその場所に留まった結果，発生した滞留点候補であり，かつ，ユーザの目的を達成するための訪問 POI poi_n を持つ．本稿では特に断りがない限り，滞留点を意図した滞留点の意味で用いる．

定義 5 (意図しない滞留点). 意図しない滞留点は，滞留点のうち意図した滞留点でないもの．ユーザが自分の意思以外で同じ場所に留まったことによって生まれた滞留点である．意図しない滞留点は訪問 POI を持たない．

定義 5 (滞留点・訪問 POI 集合). 滞留点・訪問 POI 集合 $\mathbf{V} = \{(\mathrm{sp}_n, \mathrm{poi}_n)\}_{n=1}^{M}$ は M 個の訪問 POI と対応する滞留点によって表現される．

4. 予備分析

本章では分析に用いたデータセットの説明を行い，予備分析を通じて (1) 既存研究において移動履歴からの行動分析に用いられてきた滞留点抽出手法の単純な適用では，高精度な訪問 POI 推定実現が困難であること，(2) たとえ真の滞留点が抽出できた場合においても訪問 POI 推定が困難な課題であること，の 2 点を検証する．

4.1 データセット

本実験では，被験者 1 人によって 16 日間 Nexus 7 による移動履歴データの収集を行った．測位には，Android OS の location クラスを用いて GPS 測位と WiFi 測位を常時行い，3 秒毎に精度が最も良い測位結果を採用した．収集した移動履歴データに対して被験者自身が各滞留点の開始時刻と終了時刻，および当該滞留点における訪問 POI のアノテーションを行った．訪問 POI の候補集合には，Foursquare Search Venues API[*7] を利用し，近傍 $R = 500[m]$ 以内の POI 集合を取得し，これを用いた．自宅などの Foursquare に未登録 POI については登録 POI と同等の情報を持つカスタム POI を作成し，データセットに含めた．結果，合計 98 個の滞留点に対して 24 種類の訪問 POI が付与されている．表 1 に，データセット中の頻出カテゴリの頻度と異なり POI 数，滞在時間の平均と標準偏差を示す．ここでは一部のカテゴリの統計値を示しており，最終行に全てのカテ

*7 https://developer.foursquare.com/docs/venues/search

表 2 滞留点抽出アルゴリズムの各パラメータにおける抽出結果 (太字: 最大値，下線:最小値)

θ_{dist}	θ_{time}	Precision	Recall	TP	FP	FN
100	1800	0.857	0.367	36	6	62
100	900	0.854	0.418	41	7	57
100	180	0.541	**0.867**	**85**	72	13
200	1800	**0.889**	0.327	32	4	**66**
200	900	0.870	0.408	40	6	58
200	180	0.447	0.735	72	89	26
500	1800	**0.889**	0.327	32	4	**66**
500	900	0.811	0.439	43	10	55
500	180	0.380	0.612	60	**98**	38

表 3 実験データに含まれる真の滞留点の滞留時間分布

Range [min.]	Freq.	Ratio	Cum. ratio
$t < 3$	4	0.04	0.04
$3 \leq t < 5$	15	0.15	0.19
$5 \leq t < 10$	24	0.24	0.44
$10 \leq t < 30$	10	0.10	0.54
$30 \leq t < 60$	5	0.05	0.59
$60 \leq t$	40	0.41	1.00

ゴリの統計値を示す．

4.2 滞留点抽出アルゴリズムの分析

本節では既存の滞留点抽出手法を単純に適用するだけでは，行動分析のための滞留点抽出が適切に行えないことを経験的に示す．既存の滞留点抽出手法においては，距離と時間に対する閾値パラメータや，Mean-Shift を用いる場合にはカーネル幅など，アルゴリズムの挙動を変更するパラメータが存在し，問題に合わせてこのパラメータチューニングを行うことで滞留点の網羅性，適合性の調整を行う必要がある．実際に，いくつかのパラメータを選択した際に，滞留点抽出結果がどのように変化し，1 つの固定されたパラメータでは適合性，網羅性の両方を担保することが困難であることを確認する．

本稿では，滞留点抽出手法としては，Zheng ら [12] の方法を選択し，これを分析に用いる．(1) 広く利用されている手法である，(2) パラメータの意味が直感的に理解しやすい，の 2 つの理由により Zheng らの方法を採用した．Zheng らの方法では，時間閾値 θ_{time} と距離閾値 θ_{dist} をあらかじめ設定し，θ_{time} 以上の期間，半径 θ_{dist} 以内に連続する測位点集合が含まれる場合に滞留点として抽出する．なお，Zheng らは文献 [11] において $\theta_{time} = 1800$[sec.]，$\theta_{dist} = 200$[m] が妥当であると述べている．

図 2 に移動履歴からの滞留点抽出の正解と誤りについて示す．この例においては，移動履歴において，2 つの訪問 POI が存在する．高精度な訪問 POI 推定のためには，訪問 POI が存在する位置の近くで滞留点抽出を行う必要がある．この例では，図の左側の訪問 POI については適切な滞留点抽出を行っている (TP; True Positive)．当該滞留点の右下には訪問 POI が存在しないにも関わらず滞留点が抽出されている．訪問 POI が存在しないため，滞留点として適切ではない測位点集合を滞留点と誤って判定している．これを本稿では抽出誤り (FP; False Positive) と呼ぶ．一方で，その右側には訪問 POI が存在し，滞留点として抽出するべき区間に対して滞留点抽出を適切に行っていない区間が存在する．このような誤りを本稿では抽出漏れ (FN; False Negative) と呼ぶ．

表 4 滞留点中心から訪問 POI の距離と平均順位

Avg. rank	Avg. distance [m]	Top-ranked ratio
4.47 ± 7.03	32.57 ± 15.42	0.102 (10/98)

これらの数字を用いて，滞留点抽出の適合率 (precision) と再現率 (recall) を以下に従って計算する:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}.$$

Zheng らの方法を用いて $\theta_{dist} \in \{100, 200, 500\}$ と $\theta_{time} \in \{180, 900, 1800\}$ のパラメータ組み合わせにおける滞留点抽出精度の評価を行った．真の滞留点の中心座標と抽出された滞留点の中心座標の距離が 50m 以下の場合に TP と判定した．結果を表 2 に示す．表において，太字が各カラムの最大値，下線が各カラムの最小値を表す．

表 2 の結果から，$\theta_{dist} = 200$，$\theta_{time} = 1800$ では，抽出された滞留点のうち約 89%が正しい滞留点ではあるものの，約 67%の滞留点については抽出漏れをしていることがわかる．一方で $\theta_{dist} = 100$，$\theta_{time} = 180$ においては，約 87%の真の滞留点を抽出し，FN の数が最小であるものの，結果的に FP の数が大きくなり，適合率としては低い値を示している．

より詳細に分析するため，データセットに含まれる人手付与の滞留点の滞留時間分布について分析を行った．表 3 に滞留時間 t の頻度分布，全体における比率，累積比率を示す．これより，約 54%の滞留点は滞留時間が 30 分未満であり，$\theta_{dist} = 200$，$\theta_{time} = 1800$ では，アルゴリズム上これらの滞留点の抽出することができないことがわかる．また，これらの滞留点を抽出するため θ_{time} の値を小さく設定すると，今度は FP が増えてしまい，適合率を低下させる問題が発生する．分析の結果，1 つの固定されたパラメータでは適合性，網羅性の両方を担保することが困難であり，また，問題に合わせたパラメータ調整が必要となることがわかった．

ここで，表 2 における再現率の値が訪問 POI 推定精度の最大値となることに注意されたい．FN 誤りが発生すると，そもそも訪問 POI 推定を行う対象が抽出されず，FP 誤りが発生すると訪問 POI が存在しない滞留点に対して訪問 POI 推定を試みるため，いずれの滞留点抽出誤りも訪問 POI 推定精度低下に直結する．そのため，我々は滞留点の網羅性を担保するために FP を許容して滞留点抽出を行

い，その結果から，滞留点判定と訪問 POI 推定を同時に行うことで，訪問 POI を考慮して滞留点が妥当であるか判定を行うアプローチを採用する．

4.3 滞留点と訪問 POI の分析

本稿で目的とする訪問 POI 推定が困難であることを検証するため，評価実験で用いるデータセットを用いて，訪問 POI の分析を行った．データセット作成の際に，あらかじめ滞留点中心座標近傍の POI を取得しており，1 滞留点あたり平均 73.3 件の訪問 POI 候補が存在する．

各滞留点について，取得した訪問 POI 候補集合の各 POI について，滞留点の中心座標からの距離を計算する．訪問 POI の順位の平均と標準偏差，滞留点中心から訪問 POI への距離の平均と標準偏差，最近傍に訪問 POI が存在する割合を表 4 に示す．

表 4 より，98 件中わずか 10 件のみが滞留点の中心座標から最近傍に真の訪問 POI が存在している．これより滞留点の中心座標最近傍の POI を推定するだけでは，高精度な訪問 POI 推定が実現できないことがいえる．

真の訪問 POI の順位は平均 4.47 であり，また，今回作成したデータセットにおいては，真の訪問 POI が滞留点の中心座標から最も離れた場合において 79.56m であり，最大順位は 63 位であった．これより，真の訪問 POI は最近傍には存在しないものの，滞留点の中心座標から近傍に存在する POI 集合に含まれていることがわかる．

訪問 POI が滞留点の最近傍に存在しない理由としては (1) 測位誤差による滞留点の中心座標の誤り，(2) サービスに登録されている POI の座標位置が正確ではない，という 2 つの理由が考えられる．(2) の解決方法としては，たとえば西田ら [14] が提案した，ユーザの訪問履歴を利用した POI の位置修正手法などを用いて，ある程度の解消は可能であると考えられる．実際，西田らの実験において位置修正を行うことで最近傍 POI を訪問 POI と予測する手法の推定精度は向上しているものの，十分な精度を達成しているとはいえない．今回の予備分析を通じて，真の滞留点が抽出された場合においても，訪問 POI が最近傍に存在する割合は 1 割程度とかなり低い値になっている．これらの結果から，抽出された滞留点に対する訪問 POI 推定も困難な課題であることがわかる．

5. 滞留点・訪問 POI 同時推定手法

本章では滞留点抽出と訪問 POI 推定を同時に行う提案手法の詳細を述べる．4.2 節で述べたように，提案手法の枠組みでは，あらかじめ誤りを許容した滞留点の候補集合とそれに対応する訪問 POI 候補集合を入力として受け取り，各滞留点候補が滞留点として適切であるかという判定と，当該滞留点における訪問 POI 推定を行う．この際，滞留点と判定されない場合には訪問 POI 推定結果は出力しない．4 節で述べたように，滞留点抽出の FP 誤りを減少させることができれば，訪問 POI 推定の抜本的な精度向上が可能になると考えられる．

また，滞留点候補が訪問 POI を持つ滞留点として適切であるかの判定には，滞留点候補から得られる情報のみならず，推定された訪問 POI の情報も同時に考慮して判定することを考える．すなわち「この場所では，長時間滞在する傾向のレストランに滞在している可能性が高いため，短い滞在時間の滞留点候補は抽出誤りである」のような，滞留点と訪問 POI の間に存在する関係を推定結果に反映させる手法の実現を試みる．

たとえば，滞留点候補を採用するかどうかの 2 値の離散変数を滞留点候補の数だけ用意し，各滞留点候補において訪問 POI 集合の数だけ状態を取る多値の離散変数を滞留点候補の数だけ用意する．それに加えて，これらの変数によって表現される目的関数を設計することで，各滞留点候補を採用した場合，採用しなかった場合における目的関数の計算が可能となり，目的関数を最大化する変数の組み合わせを求める組み合わせ最適化問題として定式化することで，同時推定が可能となる．しかしながら，滞留点候補数や訪問 POI 候補数が多くなると，離散変数の取りうる値の組み合わせ数が指数爆発を起こし，目的関数を最大化する解，すなわち厳密解の求解が NP 困難となり，実用的な時間での求解が困難となる．

そこで我々はこの滞留点抽出と訪問 POI 推定を同時に行う組み合わせ最適化問題を整数計画問題として定式化し，汎用ソルバを用いて求解することで，実用的な時間で解を求める同時推定手法を提案する．近年，汎用ソルバの性能向上により多くの組み合わせ最適化問題の厳密解を実用的な時間で求められるようになってきている．そこで，本稿においては高精度な滞留点判定および訪問 POI 推定を実現するために目的関数の設計を工夫し，滞留点・訪問 POI 同時推定を整数計画問題として定式化し，定式化した問題を汎用ソルバを用いて求解することで最終的な滞留点・訪問 POI 推定結果を取得する方法を提案する．

5.1 同時推定手法の整数計画問題としての定式化

図 3 に同時推定法の概略図を示す．この例では 5 つの滞留点候補が抽出された際の定式化を示している．各滞留点候補における訪問 POI 候補に対応する変数，そして，滞留点間の POI から POI への遷移を表現する変数を用意する．本稿では 0-1 整数計画問題として定式化するため，特に断りがなければ変数は 0-1 変数とする．

定式化に必要な記号と記号の説明一覧について表 5 に示す．滞留点候補 i を滞留点として採用する場合には $s_i = 1$，そうでない場合には $s_i = 0$ となるような変数と，滞留点 i において POI k を訪問すると判定した場合には $x_{ik} = 1$，そうでない場合には $x_{ik} = 0$ となる変数，滞留点候補 i に

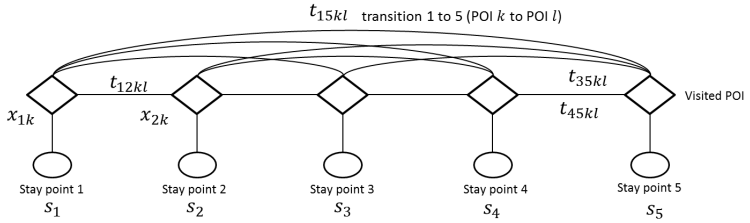

図 3 同時推手法における各変数の意味

表 5 記号の説明

記号	概要
$s_i \in \{0,1\}$	滞留点候補 i が滞留点である ($s_i = 1$) か，滞留点ではない ($s_i = 0$) ことを表す変数
$x_{ik} \in \{0,1\}$	滞留点候補 i における訪問 POI が k である ($x_{ik} = 1$) か，そうでない ($x_{ik} = 0$) ことを表す変数
$t_{ijkl} \in \{0,1\}$	滞留点候補 i において訪問 POI が k であり，滞留点候補 j における訪問 POI が l である ($t_{ijkl} = 1$)．
$y_m \in \{0,1\}$	系列における訪問回数が m の場合 $y_m = 1$，そうでない場合 $y_m = 0$
p_i	滞留点候補 i の滞留点らしさの重み
q_{ik}	滞留点候補 i に対する訪問 POI 候補 k のもっともらしさの推定値
r_{kl}	訪問 POI k を訪れた後に l を訪れるもっともらしさの推定値
α_m	セッション内の滞留点数 m のもっともらしさの推定値

おいて訪問 POI は k であり，滞留点候補 j において訪問 POI が l である場合に $t_{ijkl} = 1$ となり，それ以外の場合には $t_{ijkl} = 0$ となる変数，セッション内における採用された滞留点の総数が m である場合に $y_m = 1$，それ以外の場合には $y_m = 0$ となる変数を用意する．また，p_i，q_{ik}，r_{kl}，α_m については，事前に与えられたアノテーション付き GPS ログデータなどの利用によってあらかじめ計算された重み係数である．

これらを用いて以下の 0-1 整数計画問題 P として定式化する:

$$\text{maximize} \quad \sum_i p_i s_i \tag{1}$$
$$+ \sum_i \sum_k q_{ik} x_{ik} \tag{2}$$
$$+ \sum_i \sum_j \sum_k \sum_l r_{kl} t_{ijkl} \tag{3}$$
$$+ \sum_m \alpha_m y_m \tag{4}$$
$$\text{subject to} \quad \sum_k x_{ik} = 1 \ \forall i \tag{5}$$
$$\sum_k \sum_l t_{ijkl} = 1 \ \forall i, j \tag{6}$$
$$t_{ijkl} \le x_{ik},\ t_{ijkl} \le x_{jl} \ \forall i, j, k, l \tag{7}$$
$$\sum_m y_m = 1,\ \sum_m m y_m = \sum_i s_i. \tag{8}$$

目的関数は以下の 4 つのスコアから構成される: (1) 滞留点の確からしさ (式 (1))，(2) 当該滞留点数における訪問 POI のもっともらしさ (式 (2))，(3) 訪問 POI 遷移のもっともらしさ (式 (3))，(4) セッションにおける滞留点数のもっともらしさ (式 (4))．以下，各スコアについて詳細と係数の計算方法を述べる．

5.1.1 スコア 1: p_i (式 (1))

滞留点 i に対するスコア 1 は $p_i s_i$ によって計算する．ここで $s_i = 1$ のときに当該滞留点を採用し，そうでない場合には滞留点として採用しない．p_i は滞留点の確からしさを表す係数であり，たとえば滞留点に含まれる測位点集合を用いて計算される尤度などを用いることができる．

本稿では，滞留点の中心座標と，訓練データに含まれる最近傍の滞留点の中心座標の距離に基づくスコアを利用する:

$$p_i = \exp(-\gamma \parallel \mathbf{x}_i - NN_{train}(\mathbf{x}_i) \parallel_2^2). \tag{9}$$

ここで γ はガウス基底関数の分散の逆数を表すパラメータであり，値が大きいほど分散が小さく，離れた点に対して素早く減衰する．また，$NN_{train}(\cdot)$ は，訓練データに含まれる滞留点集合のうち，入力された滞留点の中心座標に最も近い中心座標を持つ滞留点を表す．

5.1.2 スコア 2: q_{ik} (式 (2))

滞留点 i に対して訪問 POI 候補 k のもっともらしさを表すスコアを加算する．q_{ik} にはたとえば西田らの方法によって推定された訪問 POI の確率値を用いることができる．制約式より $s_i = 0$ であるとき，変数 $x_{ik} = 0\,(\forall k)$ であるため，当該滞留点に対するスコアは加算されないことに注意する．

スコア 2 は (1) 訪問 POI カテゴリにおける滞在時間のもっともらしさ，(2) 滞留点数の中心と訪問 POI の距離の 2 つの観点に基づいて計算する．我々は，西田ら [14] の仮定と同様に，各 POI カテゴリがカテゴリ固有の滞在時間に関わるパラメータを持ち，訪問 POI における滞在時間はこのパラメータに従う確率分布によって発生した，という仮定を置く．訪問 POI カテゴリにおける滞在時間のもっともらしさのスコア計算には，各カテゴリが持つ滞在時間の確率分布の尤度にしたがって計算する．具体的には確率分布として西田ら [14] と同様に対数正規分布を利用し，パ

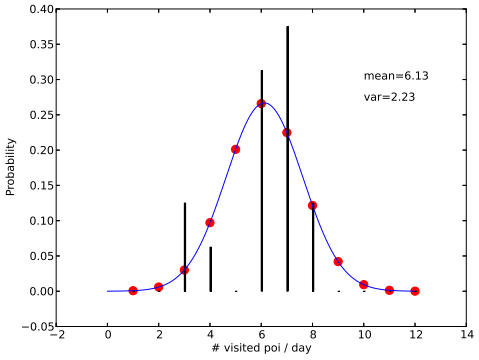

図 4 1 日あたりの訪問 POI 数の分布

ラメータの推定には最尤推定を用いる．対数正規分布パラメータの最尤推定は，

$$\nu_c = \frac{1}{|I_c|}\sum_{i\in I_c}\ln(sp_i.st), \tag{10}$$

$$\tau_c = \frac{1}{|I_c|}\sum_{i\in I_c}\{\ln(sp_i.st)-\tau_c\}^2 \tag{11}$$

によって行う．ここで $I_c \equiv \{n|cat(sp_n.vp)=c\}$ であり，カテゴリ c である POI を訪問した滞留点の添え字集合を表し，$cat(sp_n.vp)$ は滞留点 n における訪問 POI $sp_n.vp$ のカテゴリを表す．$|I_c|$ は集合の大きさを表し，カテゴリ c である POI を訪問した滞留点の総数を表す．

5.1.3 スコア 3: r_{kl} (式 (3))

訪問 POI の遷移に基づくスコアを計算する．これには観測データにおける訪問 POI カテゴリの遷移確率を用いる．今回の定式化では訪問 POI の遷移が何時点離れているかということを考慮することができない．そのため，観測データから POI k を訪問した後に POI l を訪問する確率を計算し，これを POI の遷移確率として用いる．POI の遷移ではなく，POI カテゴリ遷移を用いる．

5.1.4 スコア 4: α_m (式 (4))

スコア 1-3 では，全ての滞留点を訪問した場合にスコアが大きくなるという系列の長さに対するコストがモデル化されていない．そこで滞留点数に対するスコア項を用意する．具体的には滞留点が m 個の際に 1，そうでない場合に 0 となるような 0-1 変数 y_m によって表現する．滞留点 m のもっともらしさを表す α_m の推定には正規分布の確率密度関数を用いる．訓練データから，最尤推定を用いて正規分布の確率密度関数を推定し，1 日あたりの滞留点数のもっともらしさを，推定したパラメータにおける正規分布の尤度に基づいて計算する．図 4 に評価データすべてを用いて正規分布の最尤推定を行った結果を示す．x 軸は訪問 POI 数を表し，y 軸は訪問 POI のもっともらしさを表している．青線が最尤推定の結果得られた正規分布であり，棒グラフが実際の観測値，赤い丸が推定訪問数を表している．

Algorithm 滞留点・訪問 POI 同時推定法

Input: $\mathbf{G} = \{g_n\}_{n=1}^{N}$
Output: $\mathbf{V}$
Initialize: $\mathbf{V} \leftarrow \emptyset$
1: Partition $\mathbf{G}$ into sessions $\{\mathbf{g}_j\}_{j=1}^{M}$.
2: **FOR** j in 1 to M
3: Extract stay points S_j from $\mathbf{g}_j$.
4: **FOR** i in 1 to $|S_j|$
5: Calculate p_i for stay point i.
6: Obtain POI candidates K for stay point i.
7: Calculate q_{ik} for each POI k in K.
8: **ENDFOR**
9: Generate ILP code and solve the problem.
10: Read results V_j from the output.
11: $\mathbf{V} \leftarrow \mathbf{V} \cup V_j$
11: **ENDFOR**
12: **RETURN V**

図 5 同時推定法のアルゴリズム

5.1.5 制約式

問題 P における各制約式はそれぞれ変数を訪問 POI 推定としての整合性を取るためのものである．具体的には，滞留点に対応する訪問 POI はただひとつである (式 (5))，滞留点 i から j への遷移パターンはただ 1 つである (式 (6))，$t_{ijkl}=1$ の場合には滞留点 i において POI k を，滞留点 j においては POI l を訪問している (式 (7))．また，あるセッションにおいてはセッション長がただひとつだけ存在する (式 (8) 左)，セッション長は滞留点数の和で計算される (式 (8) 右) という意味を持っている．

5.2 アルゴリズム

提案手法の全体の処理の流れを図 5 に示す．提案手法は測位点集合 $\mathbf{G}$ を入力として受け取り，全セッションの滞留点・訪問 POI 集合 $\mathbf{V}$ を出力する．入力された測位点集合を M 個のセッションに分割し (Line 1)，各セッションについて滞留点候補の抽出を行い (Line 3) ，各滞留点について滞留点の確からしさの推定 (Line 5)，訪問 POI 候補の取得 (Line 6)，取得した候補について訪問確率の推定 (Line 7) を行う．これらを用いて整数計画問題のコードを生成し，汎用ソルバを用いて求解を行い (Line 9-10)，得られた結果を滞留点・訪問 POI 集合に追加する (Line 11)．上記の処理を全セッションについて実行する．

6. 評価

評価実験では，4.1 節で述べたデータセットを用いた．提案手法においては，アノテーションデータからパラメータや遷移確率の推定を行う必要がある．そのため，評価実験では 5 分割交差検定で評価を行い，訓練データとして利用するデータからパラメータ推定を行った．ベースラインと

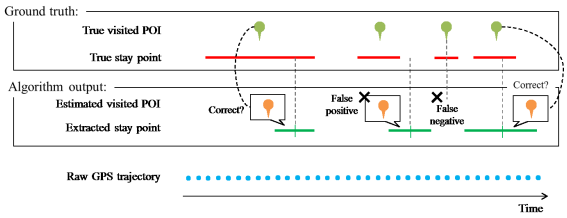

図 6 評価指標の計算方法

表 6 実験結果

θ_{dist}	θ_{time}	Method	Precision	Recall
100	180	NN	.111	.181
		JointEst	**.201**	**.308**
200	1800	NN	.105	.039
		JointEst	**.250**	**.091**

して，抽出滞留点の最近傍の POI を推定結果として選択する手法 (NN) と比較を行った．提案手法の求解に用いる ILP ソルバには Gurobi Optimizer v.5.5.0[*8]を利用した．

評価指標の計算方法について図 6 を用いて説明する．実験に利用したデータセットにおいては，被験者によって付与された正解滞留点と正解訪問 POI が存在する．図の例では，移動履歴に対して 4 件の滞留点と訪問 POI が付与されている．これに対してアルゴリズムの出力として，

今回は抽出滞留点の中心時刻 $((sp.bt + sp.et)/2)$ を時間範囲として包含する正解滞留点を正解滞留点として紐づける．図の例では，抽出した 3 つの滞留点のうち 2 件が正解滞留点と紐づいている．正解滞留点と紐づけられた抽出滞留点については，正解訪問 POI が存在するため，当該抽出滞留点に対して推定した訪問 POI との一致を判定し，一致する場合には True Positive (TP)，誤った場合には False Positive (FP) と判定する．対応する正解滞留点が存在しない場合にも FP と判定し，正解滞留点に対してどの抽出滞留点も紐づけられない場合には False Negative (FN) として判定する．これらの結果から，式 (1) と同様に適合率と再現率を計算する．ただし，ここで計算する適合率と再現率の意味は表 2 におけるものと異なることに注意する．

評価結果を表 6 に示す．2 つの異なるパラメータ設定を用いた滞留点候補抽出のいずれにおいても，提案手法である JointEst が適合率，再現率ともに高い値を示した．滞留点候補中心の最近傍 POI を訪問 POI として推定する NN の結果は，予備分析における正解滞留点の最近傍が実際の訪問 POI と一致したのは 0.102 (表 4) であり，予備分析の結果と同じような結果が得られた．

7. おわりに

本稿では，GPS ログから得られたユーザ履歴から滞留点抽出と訪問 POI 推定を同時に実現する同時推定手法を提案した．提案手法においては，滞留点抽出，訪問 POI 推定を整数計画問題として定式化することにより，滞留点抽出結果を採用しなかった場合においても訪問 POI の遷移情報を考慮した同時推定が可能となる．また整数計画問題として定式化することで，あとからドメイン知識を目的関数に導入することが容易であるため，モデルの精緻化が容易であるという利点が挙げられる．

我々の提案する同時推定手法は，目的関数を規定するものではないため，既存の POI 推定手法 [14][6][9] の結果を目的関数に取り入れることも可能である．今後は既存手法との組み合わせも視野に入れて目的関数の精緻化を行い，高精度なユーザコンテキスト推定の実現を目指す予定である．

[*8] http://www.gurobi.com/

参考文献

[1] Adams, B., Phung, D. and Venkatesh, S.: Extraction of social context and application to personal multimedia exploration, *Proc. ACM Multimedia '06*, pp. 987–996 (2006).
[2] Ashbrook, D. and Starner, T.: Using GPS to learn significant locations and predict movement across multiple users, *Personal Ubiquitous Comput.*, Vol. 7, No. 5, pp. 275–286 (2003).
[3] Cheng, Y.: Mean Shift, Mode Seeking, and Clustering, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 17, No. 8, pp. 790–799 (1995).
[4] Ester, M., Kriegel, H.-P., Sander, J. and Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Proc. KDD '96*, pp. 226–231 (1996).
[5] Kang, J. H., Welbourne, W., Stewart, B. and Borriello, G.: Extracting places from traces of locations, *Proc. WMASH '04*, pp. 110–118 (2004).
[6] Lian, D. and Xie, X.: Learning location naming from user check-in histories, *Proc. GIS SIGSPATIAL '11*, pp. 112–121 (2011).
[7] Liu, T.-Y.: *Learning to Rank for Information Retrieval*, Springer (2011).
[8] Roth, D. and tau Yih, W.: A Linear Programming Formulation for Global Inference in Natural Language Tasks, *Proc. CoNLL '04*, pp. 1–8.
[9] Shaw, B., Shea, J., Sinha, S. and Hogue, A.: Learning to rank for spatiotemporal search, *Proc. WSDM '13*, pp. 717–726 (2013).
[10] 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).
[11] Zheng, Y., Zhang, L., Ma, Z., Xie, X. and Ma, W.-Y.: Recommending friends and locations based on individual location history, *ACM Trans. Web*, Vol. 5, No. 1, pp. 5:1–5:44 (2011).
[12] Zheng, Y., Zhang, L., Xie, X. and Ma, W.-Y.: Mining interesting locations and travel sequences from GPS trajectories, *Proc. WWW '09*, pp. 791–800 (2009).
[13] Zheng, Y. and Zhou, X.(eds.): *Computing with Spatial Trajectories*, Springer (2011).
[14] 西田京介，戸田浩之，　倉島健，　内山匡：確率的訪問 POI 分析: 時空間行動軌跡からのユーザモデリング，マルチメディア、分散、協調とモバイル (DICOMO2013) シンポジウム，pp. 334–345 (2013).