

災害復旧期の交通需要予測のための
時間変化するパラメータのリアルタイム推定手法
Real-time Estimation Method of time-varying Activity Parameters
for Traffic Demand Prediction under disaster restoration period

望月 陽介¹, 浦田 淳司²
Yosuke MOCHIZUKI¹, Junji URATA²

本研究では、災害復旧期の交通需要予測に向けたリアルタイムのパラメータ推定手法を開発した。最適化問題において計算時間を大幅に短縮可能な surrogate-based optimization をベースに、アクティビティシミュレータのパラメータ推定を行うアルゴリズムの開発と実装を行った。提案アルゴリズムでは、転移学習による機械学習モデルの更新、損失関数微分値を利用した勾配降下法、学習データの範囲制限、の三つの拡張を導入した。提案アルゴリズムの有効性は、簡易なアクティビティシミュレータにより示した。また、2016年熊本地震の状況を想定した地方都市圏を対象としたアクティビティシミュレータに、提案アルゴリズムを適用した。既存アルゴリズムに比べパラメータ推定に要する時間を大幅に短縮し、リアルタイム推定で得たパラメータはシミュレータ精度の向上に貢献した。

Keywords: 数理最適化, サロゲート最適化, アクティビティシミュレータ, 災害復旧期制御

1. はじめに

1.1 背景と目的

災害被害による道路や鉄道の交通ネットワーク被害は、混雑を発生させ、復旧期の支援活動や事業継続を困難にさせる。同時に、支援者側の物資補給、応急復旧のための対応や被災者の物資獲得行動に起因する交通需要は、平時の交通需要パターンと大きく異なり、混雑をより拡大させる。ただし、災害復旧期の交通状態や行動は、常時観測型のプローブデータの取得範囲が広がる中、観測可能になりつつある。実際、桑原ら¹⁾では、パッシブ型行動データを用い、発生直後から数週間の災害復旧期において、人々の活動パターンが刻々と変化することを明らかにした。災害復旧期の交通円滑化は、支援活動の遂行や被災生活の向上に寄与することは明らかであり、円滑化のための最適な物資配送²⁾、一般車両の流入制御、臨時公共交通の供給³⁾といった復旧期の交通施策が必要である。交通需要自体が刻々と変化する中では、交通施策も需要に対応するように動的に変化することが望ましい。そのためには、災害復旧期において、時々刻々と変化する交通需要を予測する必要がある。

浦田ら⁴⁾や佐々木ら⁵⁾では災害復旧期の交通需要予測のための交通需要シミュレータを開発し、被災者と非被災者双方の交通需要の同時予測を試み、復旧期の行動需要予測を達成した。一方で、シミュレータの予測精度は

シミュレータ内の行動モデルのパラメータに依存しており、日々の行動選好の変化を含めた予測にはリアルタイム性をもったパラメータ更新が必要である。しかし、既往研究⁴⁾では、大規模な交通需要シミュレータのパラメータ推定に1日以上かかっており、パラメータを高頻度に更新するためにはパラメータ推定手法の改良が必要である。

本研究では、サロゲート型最適化のアルゴリズムを採用したオンライン型のパラメータ更新手法を提案する。提案手法では、まず、生成したパラメータセットにおいて実行したシミュレータの出力値を機械学習モデルにより事前学習する。次に、直前の観測データを用いて、構築した事前学習モデルを更新する。最後に、更新した学習モデルを用いて、最適となるパラメータセットを探索する。事前に学習モデルを構築することで、パラメータ探索にかかる時間を短縮し、オンライン型のパラメータ更新を可能とする。また、災害復旧期の動的交通制御には、需要予測の精度向上が求められる。提案する高速なパラメータ推定手法は、より直近の観測データのパラメータへの反映を可能とするため、精度向上に貢献する。加えて本研究では、予測精度向上のため、学習対象として微分値を用いる工夫、生成するパラメータセットの範囲を限定して学習データの生成密度を高める工夫を導入する。

提案するオンライン型パラメータ更新手法は、パラメ

1 学生会員, 修士(工学), 東京大学大学院 工学系研究科社会基盤学専攻

〒113-8654 東京都文京区本郷 7-3-1 e-mail: mochizuki@bin.t.u-tokyo.ac.jp Phone: 080-5040-3663

2 正会員, 工学博士・TOP, 東京大学大学院 工学系研究科社会基盤学専攻

ータ推定に時間がかかる複数コンポーネントを組み合わせた交通需要シミュレータ全般に適用可能である。しかしながら、行動選好が刻々と変わる可能性の高い異常時・災害復旧期により有効な方法であり、本研究では災害復旧期の交通需要予測を対象とした実証を行う。

1.2 本研究の特徴

高計算コストのシミュレータを下位問題とした最適化計算を高速に行うため、機械学習分野では、Surrogate-based Optimization⁶⁾⁷⁾（サロゲート型最適化）のアルゴリズムが近年開発・実装されている。サロゲート型最適化では、まず、高計算コストのシミュレータを近似学習し、高速に目的関数を出力できるサロゲートモデルを構築する。構築したサロゲートモデルにより、最適解を探索する。サロゲートモデルの活用により、目的関数の算出のために繰り返し行う下位問題（シミュレータ計算）の計算コストを削減できる。しかし、サロゲート最適化では、一つの最適解を得ることを目的としており、オンライン型の最適解更新を想定していない。そのため、計算精度向上が必要な場合は、学習データの再サンプリングとサロゲートモデルの再構築を行う。再サンプリングにはシミュレータ実行が必要である。本研究では短時間でのパラメータ推定・更新を目指しており、予測精度向上のために再構築過程を取り入れることは望ましくない。

本研究では、サロゲート型最適化の基本フレームをベースに、予測精度向上のための拡張を加えることで、災害復旧期の交通需要予測に資するオンライン型パラメータ更新手法を提案する。具体には、次の3つの拡張を取り入れる。

1. 転移学習⁸⁾⁹⁾により機械学習モデルを更新
2. 損失関数の微分値をサロゲートモデルの学習対象とする
3. 学習データの生成範囲を限定する

各拡張の貢献を簡潔に説明する。1. により、直前の観測データを用いて、事前学習した機械学習モデルを更新することで、より直近の交通状態を考慮したモデルをパラメータ更新に利用できる。最新の観測の反映により、予測精度は向上する。また、転移学習により、短時間で観測データを学習モデルに反映することができるため、ごく短い間隔でのパラメータ更新を可能とする。2. の微分値の活用により、観測誤差やシミュレータ内部の乱数を活用した計算によるランダム性の影響を最小限に留める。目的関数の極大・極小値近傍では山・谷のピーク値を学習することになり、学習の難易度が上がるが、微分値とすることで0近傍の目的関数値を学習すればよく、情報量が増える。このため、ランダム性にも頑健な学習が可能となる。なお、最適解の探索には、探索方向を示す微分値を得ることができれば十分である。3. 学習範囲を制

限することで、学習データの密度を高め、事前学習モデルの精度を向上させる。対象とする選好パラメータは突然大きく変わることはなく、刻々と変化すると想定できる。つまり、前時間帯のパラメータ周辺を中心にパラメータセットを生成し、学習することで、精度の高い学習モデルを得られる。

2. 提案アルゴリズム

本章では、提案アルゴリズムを示す。はじめに全体のフローを示してアルゴリズムで行う操作の流れを整理した後に、個別の操作について具体的な処理を説明する。

2.1 全体の推定フロー

提案アルゴリズムの適用先では、一定時間毎に繰り返しパラメータ推定を行う状況を想定している。そのため提案手法では、機械学習モデルと推定パラメータをオンラインで更新することで、新しい推定パラメータを得る。図-2にて、第 $t-1$ 期の機械学習モデル $Model_{t-1}$ と推定パラメータ β_{t-1}^* が得られている状態から、第 t 期のパラメータ推定を行う流れを示す。なお、1期は観測値入手する時間間隔を表し、4章で行う復旧期シミュレータのパラメータ推定では1時間に相当する。

図-2に示した操作は、各期で行われる操作である。提案手法は、生成したデータ群を学習データに使い、機械学習モデルの学習を行う学習段階と、機械学習モデルの出力を用いてパラメータの更新を行う更新段階の二つで構成する。学習段階と更新段階を、観測値の入手と同じ期間で每期行うことで每期の推定パラメータを得る。本アルゴリズムは、サロゲート型最適化に1.2節で述べた拡張を施したアルゴリズムである。拡張では、繰り返しパラメータ推定を行うことのみを、パラメータ推定時の仮定としている。そのため、サロゲート型最適化と同様、採用する機械学習モデルや最適化アルゴリズムの手法に関わらず、適用可能である。

2.2 個別操作

(1) 学習段階

学習段階では、 $t-1$ 期の機械学習モデル $Model_t$ を更新する。 $t-1$ 期の機械学習モデルは、 $t-1$ 期での観測データ $observe_{t-1}$ を正解値として算出した損失関数の微分

係数 $\frac{\partial Loss(\beta, observe_{t-1})}{\partial \beta}$ を出力する。 t 期の観測データ

$observe_t$ は前期の観測データとは当然異なっており、その場合損失関数も変化する。そのため、 t 期の観測データ

を正解値とする損失関数の微分係数 $\frac{\partial Loss(\beta, observe_t)}{\partial \beta}$ を精

度良く予測するために、 $observe_t$ にあわせた機械学習モデルの更新を行う。機械学習モデルの更新は、既存のモ

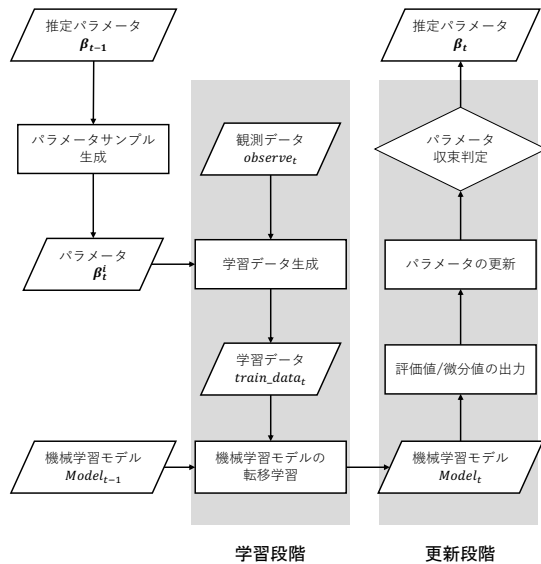


図1 提案アルゴリズムのフロー

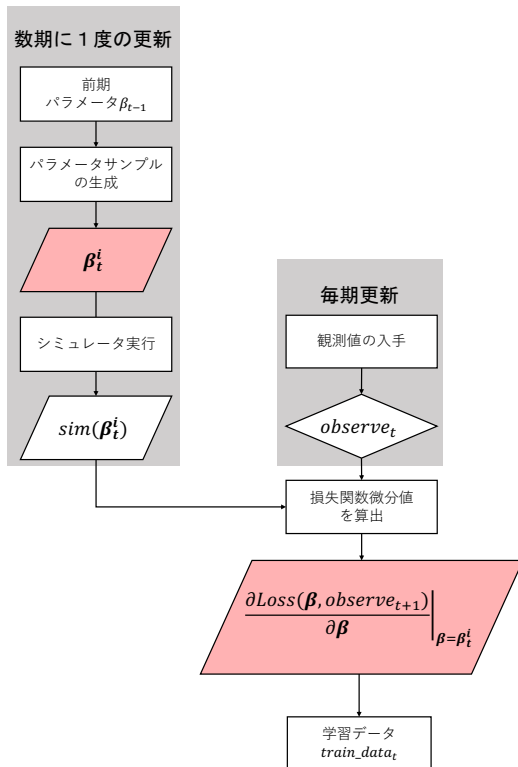


図2 学習データ作成手順

デルを新しい学習データに対して学習させる転移学習を適用する。時間的に連続して入手できる学習データに対して転移学習を行うことで、観測データの入手と並行して機械学習モデルを更新できる。

転移学習を用いたモデル更新を行うにあたり、 $Model_{t-1}$ から $Model_t$ への転移に用いる学習データ $train_data_t$ を生成する。図1で t 期の学習データ $train_data_t$ の生成フローを示した。学習データ生成にはパラメータサンプル β_t^i と、観測データ $observe_t$ が必要である。図2の左側はパラメータサンプルを生成してシ

ミュレータの実行を行う操作であり、図1ではパラメータサンプル生成に相当する。図1の右側の操作は観測データとシミュレータ実行結果から損失関数微分値を計算する操作であり、図2では学習データ生成に相当する。

本アルゴリズムでは、観測データ入手から推定パラメータ入手までの時間を短縮するために、サロゲート型最適化で一般的に行われる追加サンプルを行わない。追加サンプルを行わずとも機械学習モデルの精度を高く保つため、パラメータサンプルの生成時に学習データ密度をパラメータ推定値周りで高める。得られている前期の推定パラメータ β_t^* の周りに $t+1$ 期の真のパラメータが存在していると仮定し、パラメータサンプルを生成する範囲を制限する。学習サンプルが取りうるパラメータ範囲は t 期の推定値 β_t^* とパラメータ範囲を決定する実数 a 、 β と同次元の単位ベクトル e を用いて $[\beta_t^* - a \cdot e, \beta_t^* + a \cdot e]$ と表す。範囲内ではグリッド状にパラメータサンプルを生成する。

生成したパラメータサンプルを用いてシミュレータを実行する段階では、シミュレータ計算時間とサンプル数に比例して大きな計算時間を要する。パラメータサンプルを生成する範囲に推定パラメータの真値が入っている場合（パラメータの変化が小さい場合）、学習に用いるパラメータサンプルを変更する必要はない。そのため、パラメータサンプル生成とシミュレータ実行は每期行わずに計算を進めておき、数日に一度、学習データに用いるパラメータサンプルを変更する。一方で、損失関数の算出に用いる観測値 $observe_t$ は、観測データの入手に合わせて每期変更する必要がある。つまり、観測データを用いた学習データ生成は每期行う。

(2) 更新段階

提案手法では、パラメータ推定において微分値を用いた最適化手法である、勾配降下型の探索アルゴリズムを採用する。機械学習モデルが予測した損失関数微分値方向へパラメータの更新を繰り返すことによって、推定したい最適なパラメータに接近する。また、微分値学習による最適化を採用することにより、ノイズに頑健な推定を行うことを目的とする。交通行動シミュレータは交通行動の異質性（ランダム性）を反映するため、乱数を用いた実行により毎回結果が異なる。このランダム性は、サロゲートモデルに対しては学習データに加わるノイズとして働き、推定精度を低下させる。微分値を用いることでノイズの影響を抑えることができる。具体的な数値検証は次章で示す。なお、探索アルゴリズムとして、具体的にはステップサイズ可変のアルゴリズム（最急降下法、Adamなど）を用いる。

表 1 有効性検証のための推定条件

手法	モデル出力	モデル入力	推定手法
1)提案	損失関数微分値	パラメータ	勾配降下法
2)サロゲート	損失関数値	パラメータ	GA
3)直接学習	推定パラメータ	観測値	モデル出力

3. 数値実験による有効性検証

提案アルゴリズムを用いたパラメータ推定の数値実験を行い、提案アルゴリズムの有効性を検証する。

3.1 数値実験の目的

本研究は、前述の通り、事前学習の枠組みを導入することで短時間でのパラメータ推定を可能とし、さらに①転移学習によるモデル更新、②微分値を学習対象とする、③学習データの生成範囲の限定、のモデル拡張により予測精度の向上を図っている。本数値実験では、この三つの拡張による精度向上の効果を確認する。

提案手法の予測精度を比較するために、表 1 の通り、3つのパラメータ推定手法を設定する。手法 1)は提案手法である。手法 2)は、通常のサロゲート最適化と同様に、モデル出力を損失関数とする。微分値は出力できないため、パラメータ推定のための最適化計算には GA（遺伝的アルゴリズム）を用いる。学習にあたってのモデル入力は、提案手法と同じく、シミュレータのパラメータである。手法 3)は、深層学習モデル（DNN: Deep Neural Network）を用い、観測値を入力として、直接パラメータを出力する。直接パラメータを深層学習により出力する直感的なアプローチである。直接、得たいパラメータを算出するため、最適化計算（探索アルゴリズム）を必要としない。なお DNN は、モデル入力を受け取り、出力を返すまでに 3 層の隠れ層と ReLU 関数を通す多層パーセプトロンを用いる。

数値実験は、次の 3 つを行い、それぞれの実験目的は次の通りである。

実験 A：転移学習を用いた場合の推定精度の確認。あわせて、事前学習の際の学習データ量による精度への影響を確認。

実験 B：学習データの生成範囲の限定有無による推定精度を比較。

実験 C：学習データの生成範囲を限定した際に、真のパラメータが範囲外にある場合の推定精度への影響を確認。

実験 A は①の拡張、実験 B、C は③の拡張による効果を明らかにする。②の拡張による効果は、3つの実験を通じて、提案手法と手法 2)3)を比較することで、明らかにする。なお、事前学習による計算時間の縮減、オンライ

ン学習による精度向上は、次章の大規模シミュレータによる計算においても検証する。

3.2 数値実験の設定

本実験では、二つの離散選択モデルを組み合わせた行動シミュレータのパラメータ推定を行う。微分可能であるように簡易な行動シミュレータを用いる。簡易シミュレータは、行動の有無と目的地の、二種類の選択を離散選択モデルで記述し、メッシュの滞在人口を予測する。メッシュ m にいる人が移動を行う効用 V_{dep}^m と確率 P_{dep}^m 、メッシュ l から m に移動する効用 $V_{move}^{l,m}$ と確率 $P_{move}^{l,m}$ を離散選択ロジットモデルの式(1)、(2)で記述する。時刻 t 、任意のメッシュ i の人口 a_t^i を用いて、時刻 t 、メッシュ m の人口の予測値は式(3)に従い、 \hat{a}_{t+1}^m と計算する。 β はモデルパラメータ、 \mathbf{X} は説明変数を表す。

$$\begin{aligned} V_{dep}^m &= \beta_{dep1} \cdot X_{dep}^{m,1} + \beta_{dep2} \cdot X_{dep}^{m,2} \\ P_{dep}^m &= \exp(V_{dep}^m) / \{1 + \exp(V_{dep}^m)\} \end{aligned} \quad (1)$$

$$\begin{aligned} V_{move}^{l,m} &= \beta_{cost} \cdot X_{cost}^{l,m} + \beta_{appeal1} \cdot X_{appeal}^{m,1} \\ &\quad + \beta_{appeal2} \cdot X_{appeal}^{m,2} \end{aligned} \quad (2)$$

$$\begin{aligned} P_{move}^{l,m} &= \exp(V_{move}^{l,m}) / \sum_{i \neq m} \exp(V_{move}^{l,i}) \\ \hat{a}_{t+1}^m &= a_t^m \cdot (1 - P_{dep}^m) + \sum_{i \neq m} a_t^i \cdot P_{dep}^i \cdot P_{move}^{i,m} \end{aligned} \quad (3)$$

なお、本実験では滞在人口を観測データとして得られる状況を仮定する。観測データは、設定したパラメータの真値を用いシミュレータで計算することで生成した、滞在人口を用いる。機械学習モデルの学習データは 2 章(2)節で述べたように、前期の推定パラメータに基づいて設定した範囲でメッシュ状に生成する。学習データの数は、実験 A では $3^5, 4^5, 5^5$ 個の三通りの場合を検証し、実験 B,C では 5^5 個の学習データを用いる。

3.3 実験結果

各実験では、手法ごとに 3 期分の推定を 10 回行い、推定パラメータと真のパラメータとのマンハッタン距離を算出し、箱髭図により結果を示した（図 3,4,5）。

実験 A（図 3）では、手法 1)3)は学習データ数の増加に伴い、推定精度が向上している。この二つの手法では、今期の観測データをオンライン推定が適切に行えていることが確認できる。

実験 B（図 4）では、学習データ範囲制限をした場合が、制限なしの場合よりも、推定精度がよくなっている。また、実験 C（図 5）から、学習範囲外に真値のパラメータがある場合、1)提案手法において精度悪化が最も小さくなっている。加えて、1)提案手法による学習範囲外の

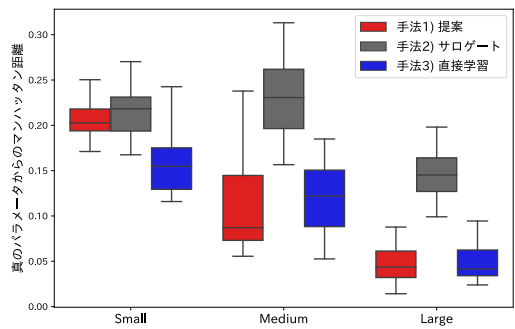


図3 オンライン学習による推定精度確認（実験A）

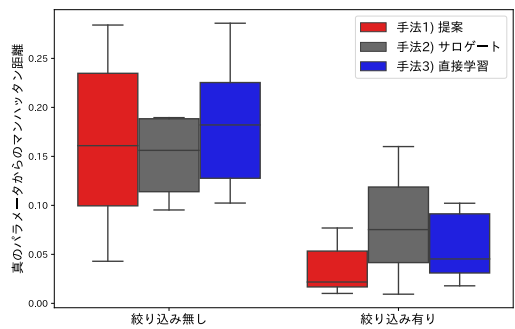


図4 学習データ範囲制限の影響比較（実験B）

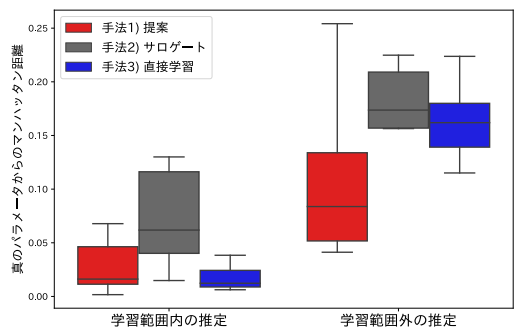


図5 学習範囲外に変化した場合の推定精度（実験C）

推定結果（実験C）は、絞り込みなし（実験B）よりも推定精度はよくなっている。つまり、たとえ、学習範囲外への真値パラメータが変化した場合でも、微分値を学習する提案手法であれば、負の影響は小さいことが確認できた。

実験A~Cを通じて、1)提案手法と3)直接学習は同程度により精度でパラメータ推定できることが確認できた。学習範囲外へのパラメータ変化があった場合は、提案手法のほうが精度がよい。ただし、大規模マルチェージェントシミュレータに提案手法を適用する場合は、微分値の学習に数値微分が必要であり、多くのパラメータを推定対象とする場合は、学習データ生成にかかるコストが大きくなる。そのため、次章の大規模シミュレーションへの適用では、提案手法と直接学習を組み合わせるパラメータ推定を行う。

4. シミュレータでのパラメータ毎時推定

本章では、2016年4月に発生した熊本地震後の熊本県をシミュレーション対象として、復旧期アクティビティシミュレータSPACE⁴⁾のパラメータ推定を行う。パラメータ推定の操作を述べた後、パラメータ推定精度を確認するための数値実験を行う。なお、実シミュレータ検証には、2016年4月の熊本県内の混雑統計データを用いる。

4.1 大規模計算における推定フローの工夫

3章の数値実験で得られた結果を踏まえて、2章で提案したアルゴリズムをパラメータ数の多い復旧期シミュレータに適用するため、4.2節の手法1と手法3を組み合わせた方法によりパラメータ推定を行う。パラメータ推定の際の、推定対象パラメータやパラメータ探索範囲、推定に用いる観測データは、元論文⁴⁾に準拠する。観測データは毎時入手できると想定し、毎時のパラメータ推定を行う。

推定対象のパラメータは12個あり、その中で特に感度の高いパラメータ $\beta_{1\sim4}$ を、1)提案手法により推定する。

表2 シミュレータでの提案アルゴリズム所要時間

操作	実行時期	所要時間
手法3 学習サンプル作成	推定開始1日目	1 サンプル 40 秒×1200 サンプル =13.3 時間
手法1 学習サンプル作成	推定開始1日目以降	1 サンプル 40~80 秒×625 サンプル =7~14 時間
手法1 推定	推定開始1日目	15.0 分
手法3 推定	最終観測値入手時	
	逆関数近似推定後	10.5 分
	毎時観測値入手時	

組み合わせた方法では、まず、少数の学習データでパラメータ推定を行える手法3) 直接学習により、全12個のパラメータを推定する（推定結果 $\hat{\beta}_{1\sim 12}$ ）。次に、感度の高い $\beta_{1\sim 4}$ に対して、より高精度な推定パラメータを毎時得るため、1)提案手法によりパラメータ推定を行う。

推定に必要な操作の所要時間を表2に示す。オンライン推定の学習サンプルの作成には7~14時間程度を要するため、学習サンプルは1日に1回更新する。一方で、オンライン推定は観測データを入手してから10分程度で行うことができ、毎時推定パラメータを得るのに十分なリアルタイム性が保証されている。学習サンプル作成の際に用いる、シミュレータ実行の計算環境は表3に示す。

4.2 シミュレータでの推定精度確認

本節では、提案アルゴリズムによるパラメータ推定精度を確認する。

推定実験では、3日間にわたり変化するパラメータを作成し、作成パラメータを入力したシミュレータ実行結果を、真の観測データとする。本実験ではパラメータ $\beta_{1\sim 4}$ を変化させ、他のパラメータは一定とした。手法3) 直接学習のみでパラメータ推定した場合と、手法1)提案手法を組み合わせてパラメータ推定を行なった場合、推定精度を比較する。

得られた推定パラメータの精度を図6に示す。パラメータ $\beta_{1\sim 4}$ の毎時推移について、真のパラメータ(黒太線)、1)提案手法で推定したパラメータ(実線)、3)機械学習モデルで直接推定したパラメータ(破線)を折れ線グラフで表した。図6は各パラメータについて、1日目の値が0で3日目の値が1になるように正規化した数値を示す。真のパラメータ値については、シミュレータが24時間分の予測を行うことを加味して、24時間の平均値をプロットしている。1)提案手法では、真のパラメータ変化を追ってオンラインに推定値を返すことができている。 β_4 のみ変化を追えていないが、推定値は近い値を返している。一方、3)直接学習による推定パラメータは、時間変化と共に徐々に変化することがなく、パラメータ真値の変化に対して感度が低いことが読み取れる。単純な機械学習モデルでのパラメータ推定に対して、提案アルゴリズムで精度高い推定を行えることを確認した。

4.3 実データによる精度検証

次に、2016年4月16日に発生した熊本地震後の二週間の滞在人口混雑統計データを用いてパラメータ推定を行い、推定結果をもとに算出した滞在人口を比較し、実現象におけるシミュレータ精度を確認する。なお、本項で用いる「混雑統計®」データは、NTTドコモが提供するアプリケーション（※）の利用者より、許諾を得た上

表3 計算環境 TokyoMarkov

GPU	Xeon(R) Gold 6128 @ 3.40GHz
OS	Ubuntu 18.04.3
メモリ	81GB
コンパイラ	g++ ver.7.4.0
オプション	-O3 -std=c++11 (-fopenmp)

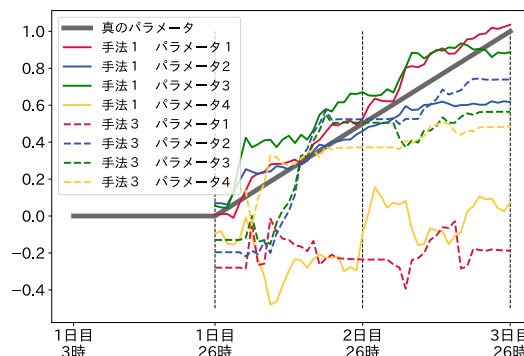


図6 シミュレータでの毎時推定パラメータの推移

で送信される携帯電話の位置情報を、NTTドコモが総体的かつ統計的に加工を行ったデータ。位置情報は最短5分毎に測位されるGPSデータ（緯度経度情報）であり、個人を特定する情報は含まれない。※ドコモ地図ナビサービス（地図アプリ・ご当地ガイド）等の一部のアプリ。

パラメータ推定と推定パラメータによるシミュレータ精度の確認を、2016年4月17日から4月24日までの観測データを用いて行う。2種類の方法でパラメータ推定し、それぞれで推定されたパラメータを用いたシミュレータ精度の比較を行う。一つの方法は1)提案アルゴリズムを採用したパラメータ推定である。観測データの入手間隔と同じ1時間毎に新規のパラメータ推定が可能であり、シミュレータでも毎時新しい推定パラメータを用いて計算を行う。比較する方法は、既存⁴⁾のGAによるパラメータ推定である。既存アルゴリズムではパラメータ推定に1日前後時間がかかることを考慮し、推定したパラメータは、入手から24時間後（翌々日）のシミュレータ計算に用いる。本実験では、GAを用いたパラメータ推定を4月16日の観測データを用いて一度だけ行う場合と、毎日の観測データを入れ替えてパラメータ推定する場合の二通りで行い、比較対象とする。

毎時のシミュレータ精度の推移を図7に示す。図では、シミュレータによる予測滞在人口と、観測された滞在人口の平均二乗誤差（RMSE）を示している。提案アルゴリズムを用いてパラメータを毎時推定した場合は、既存アルゴリズムの場合に比べて、シミュレータ精度が大きく向上していることが確認できる。4月19日以降は、提

案アルゴリズムにより推定したパラメータにより、シミュレータの予測誤差が小さくなっており、十分な再現性を確保できている。ただし、4月18日の予測ではどの場合でもRMSEは大きく、シミュレータにより記述しきれない可能性が考えられる。

以上より、提案アルゴリズムを用いて毎時パラメータを推定することにより、シミュレータ予測精度が向上することが示された。特に、GAにより毎日パラメータ推定を行ったとしても、予測精度の向上は十分ではなく、提案アルゴリズムが観測データ入手直後にパラメータ推定することの意義が明らかとなった。

5. おわりに

本研究では、大規模交通シミュレータの毎時変化するパラメータを、リアルタイムに推定するアルゴリズムを提案した。大きい計算時間を要するシミュレータに適用される Surrogate-based Optimization を元に、復旧期の観測データを用いた交通シミュレータのパラメータ推定で現れる課題を解決するような拡張を実装し、提案アルゴリズムとした。従来と比べて新たに導入した拡張を以下に示す。

1. 転移学習により機械学習モデルを更新
2. 損失関数の微分値をサロゲートモデルの学習対象とする
3. 学習データの生成範囲を限定する

これら拡張の効果を、数値実験により検証した。検証については、より一般性を高める設定で行うことで、異なる条件下での成立を確認することもできよう。シミュレータへの適用では、既存アルゴリズムから大幅に推定時間を短縮することに成功した。その結果、シミュレータ計算に用いる推定パラメータの更新がリアルタイムで可能になり、シミュレータ精度の向上を達成した。

本研究の交通分野での貢献は、本アルゴリズム開発により、従来は行えなかったリアルタイムのパラメータ推定を、大規模交通シミュレータで可能にした点にある。行動の特徴が時々刻々と変化することが明らかになっている災害復旧期において、真の行動パラメータは1時点で推定したパラメータから時間と共に離れていき、シミュレーションの精度が損なわれる。災害復旧期の施策検討において、施策効果の計算を常に正確に行うために、交通シミュレータと共に提案アルゴリズムの活用が望まれる。

謝辞

東京大学の羽藤英二教授には、アルゴリズムの発展方針について多くのご意見をいただいた。本研究の一部は、JSPS 科研費 20H00265、20K14844 の助成を受けたものです。

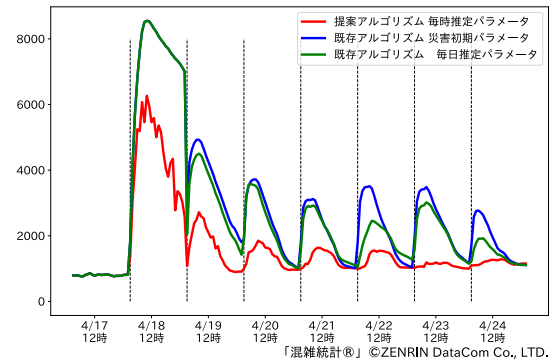


図7 シミュレータでの毎時推定パラメータの推移

参考文献

- 1) 桑原雅夫・原祐輔・三谷卓摩・川崎洋輔・竹之内篤・井料隆雅, 浦田淳司：熊本地震における都市内 交通、避難の実態と課題. 土木計画学研究・講演集, pp. 54-58, 11 2016.
- 2) 河瀬理貴, 井料隆雅, 浦田淳司：情報の不確実性を考慮した救援物資の在庫輸送戦略の数理解析, 土木学会論文集 D3（土木計画学）, 2021 年 77 巻 3 号, pp.184-200, 2021.
- 3) 神田佑亮, 井料隆雅, 藤原章正, 塚井誠人, 力石真, 三村陽一：平成 30 年 7 月豪雨時の広島～呉間の公共交通サービスの確保・向上策とその効果検証, 土木学会論文集 B1（水工学）, 2019 年 75 巻 1 号, pp. 340-349, 2019.
- 4) 浦田淳司, 佐々木泰, 井料隆雅：災害復旧期におけるアクティビティシミュレータの開発と適用. 第 61 回土木計画学研究発表会, 2020.
- 5) 佐々木泰, 浦田淳司, 井料隆雅：災害復旧期における日常活動需要計算のためのアクティビティシミュレータの構築. 第 60 回土木計画学研究発表会, 2019.
- 6) Alexander I. J. Forrester, Andr as S obester, and Andy J. Keane: Engineering design via surrogate modelling, Wiley, 2008.
- 7) Zhong-Hua Han and Ke-Shi Zhang: Surrogate-based optimization, Real-World Applications of Genetic Algorithms, pp. 343-362, 2012.
- 8) Karl Weiss, Taghi M. Khoshgoftaar, and Ding Ding Wang: A survey of transfer learning, Journal of Big Data, Vol. 3, 9, 2016.
- 9) Peilin Zhao, Steven C.H. Hoi, Jialei Wang, and Bin Li: Online transfer learning, Artificial Intelligence, Vol. 216, pp. 76-102, 2014.
- 10) Jack P.C. Kleijnen: An overview of the design and analysis of simulation experiments for sensitivity analysis,

- European Journal of Operational Research, Vol. 164, pp. 287–300, 2005.
- 11) Ding Yang, Yimin Sun, Danilo Di Stefano, Michela Turrin, and Sevil Sariyildiz: Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate- based optimization in building design, 2016 IEEE Congress on Evolutionary Computation, CEC 2016, pp. 4199–4207, 11 2016.
 - 12) Atharv Bhosekar and Marianthi Ierapetritou: Advances in surrogate based modeling, feasibility analysis, and optimization: A review, Computers and Chemical Engineering, Vol. 108, pp. 250–267, 1 2018.
 - 13) Wim C.M. Van Beers and Jack P.C. Kleijnen: Kriging interpolation in simulation: A survey, Proceedings of the 2004 Winter Simulation Conference, Vol. 1, pp. 113–120, 2004.
 - 14) Sun Hye Kim and Fani Boukouvala: Surrogate-based optimization for mixed-integer nonlinear problems, Computers and Chemical Engineering, Vol. 140, pp. 1–21, 2020.
 - 15) Atılım Güneş, Güneş Baydin, Barak A Pearlmutter, and Jeffrey Mark Siskind: Automatic differentiation in machine learning: a survey, Journal of Machine Learning Research, Vol. 18, pp. 1–43, 2018.
 - 16) Aatila Mustapha, Lachgar Mohamed, and Kartit Ali: An overview of gradient descent algorithm optimization in machine learning: Application in the ophthalmology field, Communications in Computer and Information Science, Vol. 1207 CCIS, pp. 349–359, 2020.
 - 17) 北野宏明：遺伝的アルゴリズム, 人工知能学会誌, Vol. 7, pp. 26–37, 1992.
 - 18) Luis Miguel Rios and Nikolaos V. Sahinidis: Derivative-free optimization: A review of algorithms and comparison of software implementations, Journal of Global Optimization, Vol. 56, pp. 1247–1293, 7 2013.
 - 19) Jorge Nocedal and Strephan J. Wright: Numerical Optimization, Springer New York, 2006.
 - 20) Robert Hecht-Nielsen: Theory of the backpropagation neural network, International 1989 Joint Conference on Neural Networks, pp. 593–605, 1989.
 - 21) Sebastian Ruder: An overview of gradient descent optimization algorithms, arxiv, <http://arxiv.org/abs/1609.04747>, 2016.
 - 22) Rich Caruana, Lorien Pratt, and Sebastian Thrun: Multitask learning, Machine Learning 1997, 28:1, Vol. 28, pp. 41–75, 1997.
 - 23) Chelsea Finn, Pieter Abbeel, and Sergey Levine: Model-agnostic meta-learning for fast adaptation of deep networks, Proceedings of Machine Learning Research, Vol. 70, pp. 1126–1135, 2017.
 - 24) Sinno Jialin Pan and Qiang Yang: A survey on transfer learning, IEEE Transactions on Knowledge and Data Engineering, Vol. 22, pp. 1345–1359, 2010.
 - 25) Steven C.H. Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao: Online learning: A comprehensive survey, Neurocomputing, Vol. 459, pp. 249–289, 10 2021.
 - 26) Martin Zinkevich: Online convex programming and generalized infinitesimal gradient ascent, Proceedings of the Twentieth International Conference on International Conference on Machine Learning, pp. 928–935, 2003.
 - 27) Carlos A Coello Coello: Theoretical and numerical constraint-handling techniques used with evolutionary algorithm : a survey of the state of the art, Comput. Methods Appl. Mech. Engrg., Vol. 191, pp. 1245–1287, 2002.
 - 28) Soora Rasouli and Harry Timmermans: Activity-based models of travel demand: Promises, progress and prospects, International Journal of Urban Sciences, Vol. 18, pp. 31–60, 1 2014.
 - 29) Quentin Le Lidec, Igor Kalevtykh, Ivan Laptev, Cordelia Schmid, and Justin Carpentier: Differentiable simulation for physical system identification, IEEE Robotics and Automation Letters, Vol. 6, pp. 3413–3420, 4 2021.