

都営バスのオープンデータによる渋滞検知の精度向上のための運行特性を考慮した手法の提案

畠中 希[†] 青柳 宏紀[‡] 藤田 智也[‡] 山名 早人[‡] 小口正人[†]

[†]お茶の水女子大学 〒112-8610 東京都文京区大塚 2-1-1

[‡]早稲田大学 〒169-8555 東京都新宿区大久保 3-4-1

E-mail: [†] {hatanaka.nozomi, oguchi}@is.ocha.ac.jp, [‡] {aoyagih, tfujita, yamana}@yama.info.waseda.ac.jp

あらまし 交通渋滞の解消は喫緊の課題である。渋滞情報は道路上に設置されている感知器等でリアルタイムに収集されているため感知器が設置されていない道路では渋滞を検知できない。他方で路線バスは地域住民の移動手段であるため、感知器がない道路も走行している。よって、本研究では網羅的に渋滞を検知するためオープンデータとして提供されている都営バスのデータを用いる。また、バスには様々な運行特性がある。例えば、バスが定刻より早く運行している際、定刻に合わせるため意図的に速度を落として運行する可能性がある。このような運行特性を考慮し特徴量の追加を行うことで渋滞検知の精度の向上を検討する。提案手法では、バスが定刻より 10 秒以上早く停留所を出発したか否かという情報を特徴量として追加した結果、F1 スコアが 0.594 から 0.606 へ向上した。

キーワード 渋滞、オープンデータ、バス、機械学習、公共交通機関

1. はじめに

交通渋滞は様々な悪影響をもたらしており、国土交通省によると日本における渋滞の経済的損失は年間 12 兆円、時間的損失は 1 人あたり年間 30 時間に上ると試算されている¹。したがって、このような損失を抑えるため渋滞情報をリアルタイムで取得し、渋滞を回避することは重要である。

従来の渋滞情報は、道路上に設置されてある車両感知器や光ビーコン等から収集されている。しかし、道路によっては感知器等が設置されていない場合もあるため、全ての道路が網羅されている訳ではない。

他方で、公共交通機関の 1 つである路線バスは、地域住民の移動手段であるため、主要でない道路を走行する場合もある。そのため、感知器等がない道路も走行している。また、近年では公共交通機関に関するデータがオープンデータとして公開されており、路線バスのデータも含まれている。

さらにバスは定刻より早く運行している際、定刻に合わせるため速度を落として運行するなどの運行特性があると考えられる。よって、本研究では、路線バスの 1 種である都営バスのオープンデータを用いて、運行特性を考慮した渋滞検知手法を提案する。

2. 関連研究

2.1. プローブ車両を用いた交通速度推定

Samal ら[1]は、2017 年、バスのリアルタイム GPS 軌道データを利用した交通速度推定を行う手法について提案した。一般に、バスの運行頻度が少ない時間帯は速度推定の精度が低くなるという問題点がある。そ

こで、Samal らは、気象データ等の外部要因のデータを利用することにより精度の向上を目指した。k-mean 法により類似した運行データについてクラスタリングを行い、予測精度を改善した。交通速度の平均二乗偏差が、クラスタリングを適用しない場合は毎時 4.0~4.5 マイルであったのに対し、クラスタリング適用後は毎時 2.9~3.3 マイルとなることを示した。

また、Kyaw ら[2]は、バスの GPS データに加え、道路の混雑要因となる道路付近のレストランやショッピングモールといった POI(Place of Interests)の数を特徴量として追加し交通速度の推定手法を提案した。

Gu ら[3]は、2020 年バスの GPS 軌道データと交通 IC カードから得られるデータを利用した交通速度を推定する手法を提案した。Gu らは、バスの停留所での停車時間、バスの平均速度、乗客のバス停留所間の移動時間という 3 つを入力としたニューラルネットワークを用いる手法を提案した。本手法によって、推定した交通速度と交通速度を代表するタクシー速度との相関係数が 0.94 となり、両者の間に高い相関があることを示した。

2.2. プローブ車両の GPS データを用いた渋滞検知

Xu ら[4]は 2012 年、複数台のバスの平均移動時間を用いた渋滞検知手法を提案した。T 分間に通過したバスの平均移動時間を算出する「T-window Average」と通過した直近の N 台のバスの走行時間を記録し、平均走行時間を算出する「N-window Average」を定義し、各々が個別に設定した閾値を超えた場合に、渋滞が発生することをシミュレーションにて示した。

¹ <https://www.mlit.go.jp/road/ir/ir-perform/h18/07.pdf>

Wang ら[5]は 2013 年, 1 台のプローブ車両から, 平均速度を位置毎に離散化し特徴ベクトルを用い, 機械学習にて交通状態を分類する手法を提案した. 学習アルゴリズムとしてランダムフォレスト, AdaBoost, SVM を用いて行い, それぞれ 91.59%, 89.43%, 87.86% の精度で検知できることを示した.

Carli ら[6]は, 2015 年, バスの GPS プローブデータを用いた自動渋滞検知手法を提案した. バスの平均速度, バスが閾値以上の速度で走行している時間の割合といった指標に着目し, 指標の異常を検知することで, 信号機故障の異常を検出した.

3. 先行研究

青柳ら[7]は都営バスのリアルタイム運行データと機械学習を組み合わせた渋滞検知手法を提案した. 渋滞検知の対象区間を池袋東口から渋谷駅東口間の明治通りとし, 停留所ごとの発車時刻のデータから走行速度を算出し, 一定の時間帯を「渋滞」と「非渋滞」とする二値分類を行った. 全停留所に対して学習器を 1 つ用意した場合の F1 スコアは 0.306, 停留所区間ごとに個別の学習器を用意した場合の F1 スコアは平均 0.399, 特定の区間だと 0.675 となった. また, 一定の時間に複数のバスが走行する場合のみのデータを使用することで精度向上を目指した. その結果, 特定の区間において F1 スコアが 0.742 へ向上した.

4. 提案手法

4.1 概要

本研究では青柳ら[7]の提案手法をもとに運行特性を考慮した特徴量を追加した手法を提案する.

一般的に渋滞検知では, 渋滞を「時速 10km 未満で走行している状態」と定義する. 本研究においても, 先述した定義を利用するため連続する 2 つの停留所の発車時刻の差異を停留所区間の距離で割ることで走行速度を求める.

また, バスの運転手は時刻表をもとに運転する. そのため, バスが停留所を定刻より早く出発した場合, 時刻表通りに運転を行おうと時間調整のため意図的に速度を落として次の停留所へ向かう可能性がある. したがって, 速度に影響を及ぼす可能性を考慮し, バスが定刻より早いかな否かを特徴量に含めることでモデルの精度向上を目指す.

4.2 特徴量抽出

提案手法で用いる記号を表 4.1 に示す

表 4.1 特徴量抽出に用いる記号

記号	定義
b_i	i 番目のバス
p_j	j 番目のバス停
s_j	p_j から p_{j+1} の区間
l_j	区間 s_j の距離[m]
td_{ij}	b_i の p_j の出発時刻
Δtd_{ij}	区間 s_j にかかる所要時間[s]
v_{ij}	b_i の s_j での速度
c_{ij}	b_i が p_j を出発する時刻を 20 分ごとに分類した指標. 0 時 0 分 0 秒から 0 時 19 分 59 秒を 0 として, 以後 20 分おきに 1, 2, 3, ... と定義する.
ttd_{ij}	b_i の p_j の出発定刻時刻
g_{ij}	td_{ij} と ttd_{ij} との差

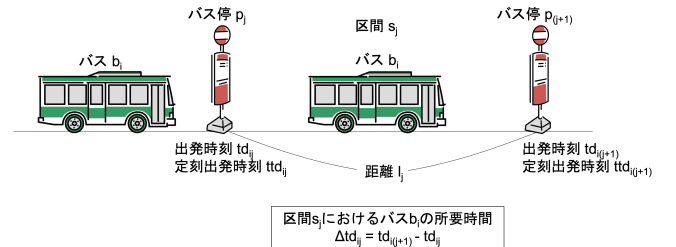


図 4.1 特徴量抽出に用いる記号の説明

本手法では, バス b_i がバス停 p_j を td_{ij} に出発したというデータを用いる. ただし, バス b_i は出発時刻順, バス停は p_j は経路順に並び替えているとする. また, 区間 s_j の距離 l_j は Google Directions API² から取得する. Δtd_{ij} は出発時刻 td_{ij} と $td_{i(j+1)}$ から (1) で表される.

$$\Delta td_{ij} = td_{i(j+1)} - td_{ij} \quad (1)$$

次に, バス b_i の区間 s_j における速度 v_{ij} を算出する. 平均速度 v_{ij} は Δtd_{ij} と l_j から次式 (2) で表される.

$$v_{ij} = \frac{l_j}{\Delta td_{ij}} \quad (2)$$

また, バス b_i のバス停 p_j の出発時刻 td_{ij} を 20 分ごとの時間帯 c_{ij} に分類する. c は, 0 時 0 分 0 秒から 0 時 19 分 59 秒を 0 とし, 以後 20 分おきに 1, 2, 3, ... と定義する.

バス b_i が停留所 p_j を出発する予定時刻を ttd_{ij} とする. この情報は公共オープンデータセンターで公開されている東京都交通局 バス停時刻表の API から取得した.

Δtd_{ij} は出発時刻 td_{ij} と ttd_{ij} から次式 (3) で表される.

$$g_{ij} = ttd_{ij} - td_{ij} \quad (3)$$

本手法で用いる特徴量を表 4.2 に示す.

表 4.2 本手法で用いる特徴量

v_{ij}	バス b_i の停留所区間 s_j 走行時の速度
$v_{(i-1)j}$	1 つ前のバス b_{i-1} の同一停留所区間 s_j 走行時の速度
c_{ij}	バス b_i の区間 s_j 走行時の時間帯
isEarly	定刻より 10 秒以上早く走行しているか

まず、上記で計算した v_{ij} , $v_{(i-1)j}$, c_{ij} の 3 つの変数の特徴量として用いる. v_{ij} , $v_{(i-1)j}$ はバスの速度であり、渋滞の定義でもあるため、特徴量として有用であると考えられる. また、 c_{ij} は時間帯を示す指標であり、時間により交通量が変化するため有用であると考えられる. さらに、isEarly は b_i が定刻より 10 秒以上早く出発しているか否かを表す. これは、出発時間が定刻との差があれば定刻に合わせるため、運行速度に影響を及ぼす可能性があり、有用であると考えられる.

4.3 分類手法

収集したバスの発車時刻のデーから表で示した 4 個の特徴量を抽出し、バス停区間 s_j において「渋滞」「非渋滞」の二値分類を行う. 学習器は、ランダムフォレスト、AdaBoost、XGBoost を用いる. ここで「渋滞」とは、時速 10km 未満で走行している状態」と定義する. 学習器は、全ての停留所区間に対して 1 つだけおく. 学習器は Python ライブラリである scikit-learn を用いて実装を行う.

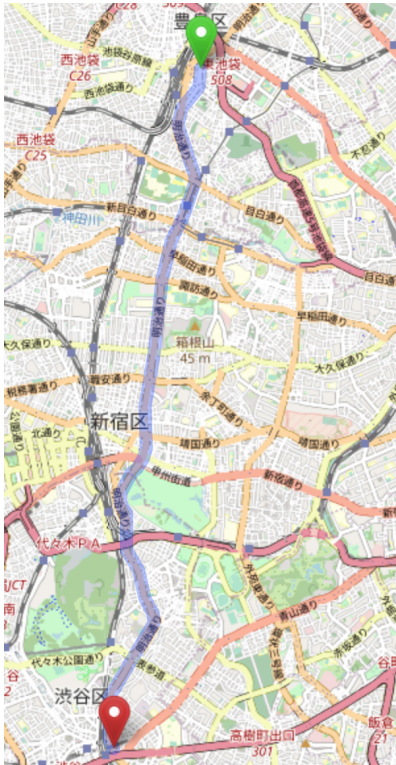


図 5.1 渋滞検知対象区間 (OpenStreetMap より)

5 評価実験

5.1 渋滞検知の対象区間

本実験で渋滞検知を行う対象区間を図 5.1 に示す

本実験で対象とする区間は、池 86 系統の「池袋駅東口」～「渋谷駅東口」～「南池袋三丁目」間とする.

また、都営バスには平日、土曜、休日の 3 種類存在するが、本実験では平日ダイヤの運行データのみをしようする.

5.2 データセット

5.2.1 バスの運行データの取得方法

都営バスの運行データは、「公共交通オープンデータセンター」の API³から取得した.

次の表 5.1 に取得したバス運行データの一部を示す. 表 5.1 における「Date」は日付、「BusNumber」はバスの運行番号、「PassPole」は出発した停留所の識別子、「PassTime」は「PassPole」の停留所を出発した時刻である.

表 5.1 取得したバス運行データ(一部)

Date	BusNumber	PassPole	PassTime
2022-11-30	C220	SodaiRiko.1051.2	10:23:49
2022-11-30	E394	JingumaeRokuchome.760.1	10:22:40
2022-11-30	E397	ShinjukuYonchome.714.8	10:21:58

5.2.2 渋滞ラベルの取得方法

渋滞ラベルの正解データは、Google Maps Platform の Directions Advanced API⁴にリクエストを送ることで取得した. データの収集期間は 2022 年 11 月 30 日～12 月 13 日の二週間と 12 月 17 日～12 月 31 日の二週間であり、5 分おきの頻度で取得した. Directions Advance API では、出発地点と到着地点を緯度経度で指定しリクエストを行うと、2 点間の道のり(m)と、リアルタイムの交通状況を踏まえた一般車両が通過する所要時間(s)を含むデータを返す. よって、バス停 b_i とバス停 b_{i+1} の緯度経度データを Directions Advance API の出発地点と到着地点に入力することにより、停留所間の距離とリアルタイムの通過所要時間を取得することができる. 取得した距離を所要時間で割ることによって、渋滞の定義で用いられるその区間での交通速度を計算できる. 計算して得られた交通速度が、渋滞の定義である時速 10km 未満であれば渋滞と判断し、そうでなければ非渋滞と判断する.

³ <https://developer-dc.odpt.org/ja/info>

⁴

<https://developers.google.com/maps/documentation/direction>

取得した渋滞ラベルデータの一部を表 5.2 に示す. 表 5.2 における「Date」はデータを取得した日付, 「Time」はデータを取得した時刻, 「From」は区間の始点停留所, 「To」は区間の終点停留所, 「Duration」は区間を一般車が通過するのにかかる時間, 「Distance」は区間の距離である.

表 5.2 取得した渋滞ラベルデータ(一部)

Date	Time	From	To	Duration[s]	Distance[m]
2022/1/30	9:0:0	池袋駅東口	南池袋一丁目	158	615
2022/1/30	9:0:0	北参道	新宿四丁目	190	1,127
2022/1/30	9:0:0	新宿四丁目	日清食品前	159	787

5.2.2 時刻表データの取得方法

時刻表のデータは, 運行データと同様に「公共交通オープンデータセンター」の API から取得した. 次の表 5.3 に取得した時刻表のデータの一部を示す. 表 5.3 における「index」は停留所を経路順に並べた際に何番目かを示す指標, 「PassPole」は停留所の識別文字列, 「departureTime」は停留所を出発した時刻を表す.

表 5.3 取得した時刻表のデータ(一部)

index	PassPole	departureTime
1	IkebukuroStationHigashiguchi.87.4	06:40
2	MinamiIkebukuroSanhome.1470.3	06:42
3	MinamiIkebukuroItchome.846.2	06:43

5.2.3 データセットの作成

本実験では「渋滞」「非渋滞」の二値分類により渋滞検知を行う. 渋滞検知に必要なデータセットの一部を表に示す. データセットはバスの運行データ(表 5.1)、渋滞ラベルデータ(表 5.2)、時刻表データ(表 5.3)を用いて作成した. 表 5.4 にデータセットの一部を示す. 表 5.4 の「Date」はデータを取得した日付, 「ToTime」は区間の始点停留所を出発した時刻, 「FromTime」は区間の終点停留所を出発した時刻, 「DifSec」は区間を通過するのにかかった時間, 「IsFast」は区間の始点停留所を定刻より 10 秒以上早く出発したか否か, 「IsJam」は渋滞か非渋滞かを示す.

本実験では 12 月の平日ダイヤ場合のみとしてデータセットを作成した.

表 5.4 作成したデータセット(一部)

Date	ToTime	FromTime	ToPassPole	FromPole	DifSec	IsEarly	IsJam
2022/12/01	10:50:58	10:56:04	Omot esando.297.3	JinNanItchome.2052.1	2.78	0	0
2022/12/01	11:04:04	11:09:14	Jingu maeRokuchome.760.2	Jingu maeItchome.758.1	2.33	0	0

6 実験

本実験では定刻より 10 秒以上早く運行するという特徴は有用であることを確認する. 用いるデータ数は 6,971 個(渋滞 1,282 個, 非渋滞 5,689 個)である. また, 学習データとテストデータの分け方は, 前半 80% を学習データ, 後半 20% をテストデータとする.

6.1 ベースライン手法

ベースライン手法として isEarly 以外の特徴量である v_{ij} , $v_{(i-1)j}$, c_{ij} を用いた場合のみで渋滞検知を行う. アルゴリズムとしてランダムフォレスト, AdaBoost, XGBoost を用いた時の accuracy, precision, recall, F1 スコアを表 6.1 に示す.

表 6.1 ベースライン手法の結果

アルゴリズム	accuracy	precision	recall	F1 スコア
ランダムフォレスト	0.869	0.784	0.497	0.594
AdaBoost	0.875	0.834	0.468	0.600
XGBoost	0.844	0.943	0.236	0.377

6.2 提案手法

ベースライン手法で用いた特徴量に isEarly を追加して渋滞検知を行う. アルゴリズムとしてランダムフォレスト, AdaBoost, XGBoost を用いた時の accuracy, precision, recall, F1 スコアを表 6.2 に示す. 最も F1 スコアが高いアルゴリズムであるランダムフォレスト

表 6.2 特徴量に isEarly を追加した結果

アルゴリズム	accuracy	precision	recall	F1 スコア
ランダムフォレスト	0.870	0.777	0.496	0.606
AdaBoost	0.874	0.829	0.468	0.598
XGBoost	0.859	0.761	0.432	0.551

の F1 スコアはベースライン手法と比較すると 0.594 から 0.606 へ向上した. 定刻より 10 秒以上早く出発したか否かを特徴量に入れることはモデルの精度向上に寄与したことが分かる.

7 まとめ

本研究では, 都営バスのリアルタイム運行データを用いた渋滞検知の精度向上のため運行特性を特徴量に追加する手法を提案した. バスが定刻より 10 秒以上早く出発したか否かを特徴量に追加することで F1 スコアが 0.594 から 0.606 へ向上した.

今後の課題としては, バスが定刻より遅く出発した場合に関しても考慮することが挙げられる.

謝 辞

本研究は, 令和 4 年度「東京都と大学との共同事業」の一環として実施した. また, 一部, JST CREST JPMJCR22M2 の支援を受けたものである.

参 考 文 献

- [1] C. Samal, F. Sun and A. Dubey, "SpeedPro: A Predictive Multi-Model Approach for Urban Traffic Speed Estimation," 2017 IEEE International Conference on Smart Computing (SMARTCOMP), 2017, pp.1-6, doi: 10.1109/SMARTCOMP.2017.7947048.
- [2] Y. Kyaw, N. N. Oo and W. Zaw, "Estimating Travel Speed of Yangon Road Network Using GPS Data and Machine Learning Techniques," 2018 15th International Conference on Electrical Engineering/ Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON), 2018, pp. 102-105, doi: 10.1109/ECTICon.2018.8619908.
- [3] Y. Gu, Y. Wang, and S. Dong, "Public Traffic Congestion Estimation Using an Artificial Neural Network," ISPRS International Journal of Geo - Information, vol. 9, no. 3, p. 152, 2020, pp.1-17, doi:10.3390/ijgi9030152.
- [4] Y. Xu, Y. Wu, J. Xu, and L. Sun, "Efficient Detection Scheme for Urban Traffic Congestion Using Buses," 2012 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 287-293, doi: 10.1109/WAINA.2012.62.
- [5] C. Wang and H. Tsai, "Detecting urban traffic congestion with single vehicle," 2013 International Conference on Connected Vehicles and Expo (ICCVE), 2013, pp. 233-240, doi: 10.1109/ICCVE.2013.6799799.
- [6] R. Carli, M. Dotoli, N. Epicoco, B. Angelico and A. Vinciullo, "Automated evaluation of urban traffic congestion using bus as a probe," IEEE Int. Conf. on Automation Science and Engineering (CASE), 2015, pp.967-972, doi:10.1109/CoASE.2015.7294224.
- [7] 青柳宏紀, 岡田一洸, 山名早人. 都バスのリアルタイム運行データを用いた渋滞検知. DEIM2022 第 14 回データ工学と情報マネジメントに関するフォーラム. 2022, pp.1-8.