

オンライン在線位置データを用いた 列車遅延伝播の予測アルゴリズムの提案

Prediction Algorithms for Delay Propagation Using Online Train Position Data

情報工学専攻 春田 雅也
Information and System Engineering HARUTA Masaya

概要: 本研究では東京メトロと都営地下鉄を対象とし、オンライン在線位置データから任意の時点の遅延状況を予測することを目標とする。まず、各事業者からオンラインに提供されるリアルタイムの在線位置データと時刻表データから遅延時間の計算および扱い方について説明する。次に、この時間経過によって変化する鉄道網の遅延状況という時系列データを扱うための予測モデルの構築や誤差評価の説明を行い、学習過程の比較および遅延伝播の予測精度の評価を実施し考察を行う。

キーワード: 公共交通機関, 機械学習, ニューラルネットワーク, ビッグデータ

1. はじめに

我が国の都市圏の公共交通網は、今日では非常に複雑な高密度ネットワークとなっており、地域住民のモビリティ変化に多大な影響を与えている。このことは、多くの選択可能な経路計画と、有事の際の代替移動手段の多様化をもたらす一方で、混雑や渋滞が常態化するなど課題も多く抱えている。特に鉄道網では、過密ダイヤや相互直通運転の路線増加により、ある地点で発生した遅延が広範囲に波及し、回復までに長い時間がかかる問題が発生しやすくなる。これらの要因により、遅延やその伝播が利用客と運営会社の双方にとって経済的損失をもたらすことが知られている [1]。

本研究では都市圏の公共交通機関としてこの鉄道網に着目し、現在時点までの遅延情報を入力として任意時点の遅延状況を予測することを目的としている。

予測と分析にあたり、入力として時間分解能の高い鉄道の運行情報に関するデータと、大量のデータから遅延状況の予測を高速に計算できるモデルが必要となる。

2. 使用データ

2.1. 提供事業者と対象路線

在線位置データを提供している事業者のうち、本研究で用いる事業者は以下の2つである。

- 東京メトロオープンデータ事務局 [2]

- 公共交通オープンデータ協議会 [3]

本研究ではこのうちリアルタイムで入手できる在線位置データと列車時刻表データを用いる。東京メトロ全線と都営地下鉄全線の在線位置データが提供されている。

2.2. 遅延時間の計算

在線位置データはリアルタイムで提供される。約30秒周期で更新され、アクセスした日時に最も近いデータが取得できる。データの時系列間隔が疎らのため、1分毎のデータにリサンプリングする。このデータは回送中である列車の位置情報は含まれていない。列車時刻表データはダイヤグラムとして計画されている列車ごとの時刻表情報であり、時間分解能は1分である。

本研究で扱う路線では時間帯ごとに旅客運行中の列車数変動する。このような場合でも、予測モデルが利用可能であることが前提となるが、列車遅延状況をそのまま扱うと任意の時刻の学習モデルへの入出力データ数が固定できない問題が生じる。したがって、列車ごとではなく駅ごとの遅延時間として再計算された遅延状況を入出力データとして扱う必要がある。取得された在線位置データと時刻表データから駅ごとの遅延時間を計算し補完するまでの手順は Algorithm1 の通りである。

2.3. 入出力データ

任意の時点 t 、現在時点 T 、予測したい時点 $F > T$ 、予測モデルに対する入力の期間 p_I 、出力の期間 p_O 、学習時のエポック数を e とする。

全路線集合を \mathcal{A} 、全駅集合を \mathcal{S} とおき、各路線 $r \in \mathcal{A}$ の駅集合を $S_r \subseteq \mathcal{S}$ とする。学習および予測対象の路線集合を $\mathcal{R} \subseteq \mathcal{A}$ 、入力に用いる駅集合を $S_I := \bigcup_{r \in \mathcal{R}} S_r$ 、出力に用いる駅集合を $S_O \subseteq S_I$ とする。さらに、ある時点における1駅分を持つデータの次元数を d とする。

時点 t の入力データ集合を \mathcal{I}_t 、出力データ集合を \mathcal{O}_t 、正解データ集合を \mathcal{L}_t とすると、このうち入出力データ集合 $\mathcal{I}_t, \mathcal{O}_t$ の各要素はニューラルネットワークにおけるニューロンとなる。

Algorithm 1 駅ごとの遅延時間を求める計算手順
 初期化: 全ての駅の遅延時間を NULL で初期化

リサンプリング後の在線位置データの中で、現在時刻に最も近い時刻を Now とする

if 列車 t が停車中 **then**

t の列車時刻表データの中から、停車中の駅の出発時間を Dep_t とする

$Now - Dep_t$ を計算し、負数であれば0としてその値を停車中の駅の遅延時間とする

else

t の列車時刻表データの中から、次駅の到着時刻を Arr_t とする

$Now - Arr_t$ を計算し、負数であれば0としてその値を次駅の遅延時間とする

end if

遅延時間が NULL となっている駅のデータを前駅の遅延時間で補完していく

3. 予測モデル

本章では、予測モデルを使った予測機構について説明したあと、予測モデルのネットワーク構造について解説する。予測モデルには3種類のネットワーク構造を用いる。

3.1. 学習機構と遅延伝播の予測機構

学習すべき予測モデルを関数 $predictionModel$ とし、入力を p_I 時点分だけのデータ集合、出力を p_O 時点分だけのデータ集合とする。

本研究では、任意時点 F までの予測を可能とするため、学習済みの予測モデルを繰り返し使用する。実際には出力を次の入力に $\lceil \frac{F-T}{p_O} \rceil$ 回フィードバックしていく。予測ベクトル計算過程のイメージを図1に示す。

3.2. Feedforward Neural Network

入力データベクトルを受け取る入力層から、任意の枚数用意された中間層を通じて、最終的な出力データベクトルを返す出力層までの流れで計算を行うネットワーク構造(以下 FNN)である。 l 番目の層に含まれるニューロンが持つ値のベクトルを z_l 、 l 番目の層から次の $l+1$ 番目の層への結合の重み行列を $W_{l,l+1}$ 、 $l+1$ 番目の層が持つ各ニューロンへのバイアスペクトルを b_l とすると、 $l+1$ 番目の層のニューロンへの入力ベクトル u_{l+1} は

$$u_{l+1} = W_{l,l+1}z_l + b_{l+1}$$

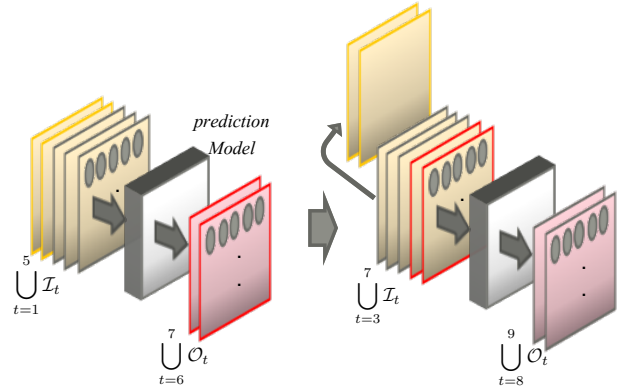


図1: 予測機構のイメージ ($p_I = 5, T = 5, p_O = 2$)

となる。出力層が L 番目であるとし、FNN のイメージを図2に示す。

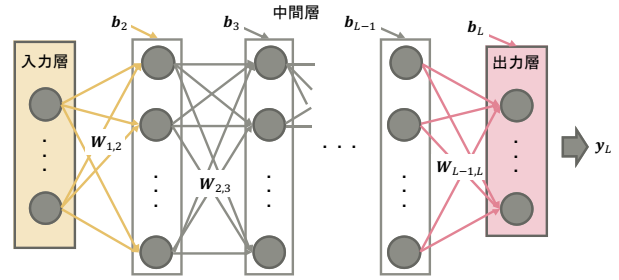


図2: FNN の構造

3.3. Extreme Learning Machine

Extreme Learning Machine[4](以下 ELM) は基盤となる構造は FNN と同じであるが、中間層が1層であり、入力層から中間層までの重み $W_{1,2}$ と中間層へのバイアスペクトル b_2 をそれぞれを最初に乱数で固定する。学習対象は中間層から出力層までの重み $W_{2,3}$ とバイアスペクトル b_3 であるネットワークである。本研究では以下の数式の通りにパラメタを固定する。 $N(\mu, \sigma^2)$ は平均値 μ 、標準偏差 σ である正規分布を表す。

$$W_{1,2} \sim N(0, 1)$$

$$b_2 \sim N(0, 1)$$

ELM の主な特徴は、ニューラルネットワークが学習によって得られる表現力を最小限に抑えつつ、計算時間を大幅に短縮しているところにある。

3.4. Recurrent Neural Network

過去の入力で計算した結果を再帰的に用いて現在時点の予測値を出力するニューラルネットワークのこと

を Recurrent Neural Network(以下 RNN) という。過去の状態を示す値を再帰的に利用できることから、時系列データなど連続的な情報を扱える利点がある。時点 t での l 番目の層が持つ値のベクトルを z_l^t とする時、 l 番目から $l+1$ 番目への計算手順は以下の通りである。

$$\begin{aligned} \mathbf{u}_l^t &= \mathbf{W}_{l-1,l}^t z_{l-1}^t + \mathbf{W}_l^{t-1,t} z_l^{t-1} + \mathbf{b}_l^t \\ z_l^t &= f(\mathbf{u}_l^t) \end{aligned}$$

ここで、同じ l 番目の層で時点 t から時点 $t+1$ への入力結合にかかる重みは $\mathbf{W}_l^{t,t+1}$ であり、同じ時点 t で l 番目の層から $l+1$ 番目の層への入力の結合にかかる重みは $\mathbf{W}_{l,l+1}^t$ 、時点 t における l 番目の層の全ニューロンにかかるバイアスベクトルは \mathbf{b}_l^t である。

本研究では中間層として LSTM ユニット [5] を用いた。また、例として中間層が 1 層である RNN の構造のイメージを図 3 に示す。

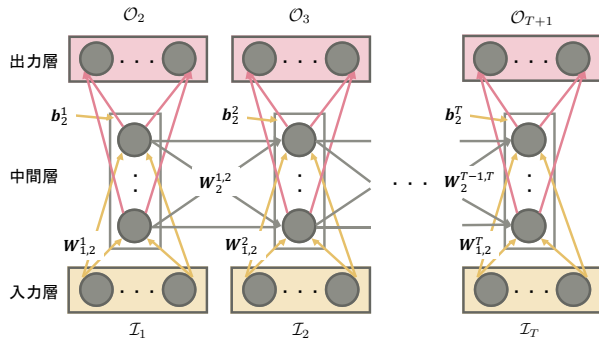


図 3: RNN の構造

3.5. 誤差関数

本研究では、遅延時間の大きさにより遅延規模をカテゴリ分類する。具体的には以下の 4 つに分類をした。これらの分類を要素として含んだカテゴリ集合を K とする。

- 遅延時間 0 分 (無遅延)
- 遅延時間 1~3 分 (小規模遅延)
- 遅延時間 4~6 分 (中規模遅延)
- 遅延時間 7~分 (大規模遅延)

p_{sk} を駅 $s \in S_O$ における遅延カテゴリ $k \in K$ の出現確率とすると、誤差評価の重みを

$$w_{sk} = -\log p_{sk}$$

とすれば、該当遅延規模の出現確率が低くなるほど重み w_{sk} が大きくなり、稀に起こる大きな遅延をより重要視して学習することが可能となる。

さらに、時点 t における出力データ集合に含まれる各要素を $o_{sd}^t \in \mathcal{O}_t$ 、正解データ集合に含まれる各要素を $l_{sd}^t \in \mathcal{L}_t$ として、誤差関数 E を

$$E = \sum_{t=T+1}^{T+p_O} \sum_{s \in S_O} w_{sk} \sum_d (o_{sd}^t - l_{sd}^t)^2 \quad (1)$$

と定める。

4. 数値実験

入力データが対象路線内の全駅である場合を all、対象路線のうち 1 路線内の全駅である場合を perDirection と表記する。さらに、出力データは対象路線内の 1 駅ごととして実験を行った。

まず、入力データ集合 \mathcal{I} が all, perDirection であるか、ネットワーク構造が FNN, ELM, RNN-LSTM であるかで合計 6 種類の数値実験を行った結果を示す。図 4 は全 6 種類の数値実験における学習過程のグラフであり、縦軸は評価関数 V の値を表し、値が小さい方がより良いモデルであることを意味している。横軸はエポック数を表し、右へ行くほど学習が進んでいることを表している。

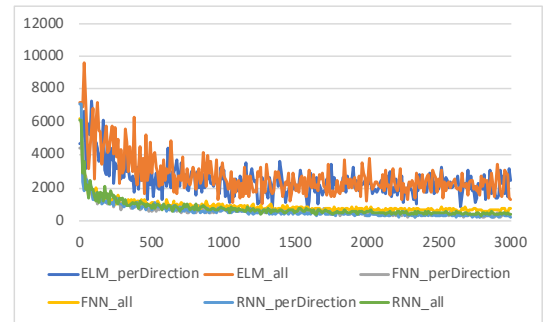


図 4: 各数値実験における評価関数値推移

ELM のネットワーク構造を用いた場合には入力データによらず学習がうまく収束しなかった。また、他のモデルと比べて学習終了時点での評価関数値も高くなっている。その他の実験設定では評価関数値の収束の速さと、学習終了時点での大きさは同程度であった。平均的に最も予測精度が良くなったモデルは RNN_all であった。

さらに、全種類の数値実験での各時点における正解データと予測された出力データの誤差が 1 分以上であった比率を、図 5 に示す。1 エポックごとに、予測した遅延時間の数のうち 1 分以上の誤差となっている比率が縦軸である。

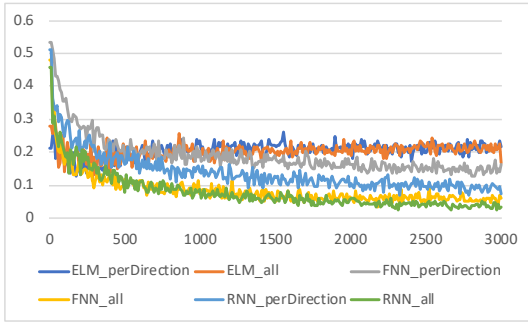


図 5: 各数値実験における 1 分以上誤差の推移

ELM ネットワーク構造を用いた予測モデルでは学習が進んでもこの誤差率が減ることはなかった。他の実験設定に関しては、入力データを全路線にした場合の方が誤差率が低く、FNN よりも RNN のネットワーク構造の方が低かった。

さらに、1 エポックごとの学習にかかる計算時間を表 1 に示す。実験時間は 2.2 節で説明した計算も含まれており、表中の値の単位は全て秒である。

表 1: 1 エポックでの予測値計算時間の比較

モデル	all	perDirection
ELM	4.43	5.47
FNN	4.46	5.51
RNN	5.07	6.27

図 6 は、式 (1) で用いられている遅延カテゴリによる重み w_{sk} を用いた場合とそうでない場合の 1 分以上誤差である比率を縦軸の値として表した比較実験の結果グラフである。予測モデルは前数値実験で最終的な評価関数値が最小となった RNN_all を用いる。

遅延カテゴリによる重みを使用の方が学習が進んでいくと誤差率が減少している。30 秒以上誤差や 2 分以上誤差でも同様の数値実験を行ったところ、同じ傾向が見られたが、2 分以上誤差ではカテゴリの使用による大きな差は見られなかった。

5. まとめ

ELM は計算時間が最も短いが学習の収束が最も遅く、急速に変化するような交通網に対応させるには不向きであることがわかる。一方 FNN と RNN では今回の実験設定では計算時間に大きな差は見られないが、RNN の方がより早く学習の収束が見られ遅延予測に対するネットワークの表現力が最も高いと考えられる。

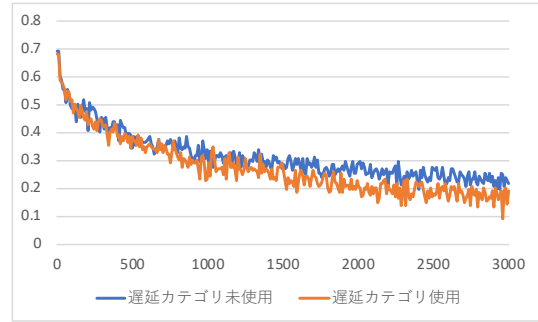


図 6: 遅延カテゴリ使用有無による 1 分以上誤差の推移

ネットワークモデルを RNN として、学習済みの予測モデルのテストをエポック数を 300 として実行したが、全体として小さい遅延時間を予測値として出力してしまい、大幅な遅延が発生している場合の誤差率が非常に高くなってしまう結果となった。出現率の低い遅延時間の予測精度を上げるためのモデル構築の更なる工夫が必要となる。

今後の課題としては、入出力データを作成するためのデータ整形にかかる計算や補完のための時間の短縮、駅毎の遅延時間計算時の欠損値補完の荒さの改善などが挙げられる。

今後の研究展望としては、新たなリアルタイムの在線位置データの提供によって相互直通運転になっている路線間の遅延伝播の学習および分析が可能であると考えられる。さらに、鉄道網ではなく、バスや飛行機など運行ダイヤがあらかじめ組み込まれている他の公共交通機関への応用なども考えられる。

参考文献

- [1] 武藤・柴田 (2009). 輸送障害に遭遇した旅客の経済損失評価法, 鉄道総研報告 23(8), 35-40.
- [2] 東京メトロオープンデータ開発者サイト. <https://developer.tokyometroapp.jp>, (2019/12/24 アクセス)
- [3] 公共交通オープンデータセンター 開発者サイト. <https://developer.odpt.org/>, (2019/12/24 アクセス)
- [4] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew (2006). Extreme Learning Machine: Theory and Applications, Neurocomputing 70, 489-501.
- [5] Gers, F. A., Schmidhuber, J. (2000). Recurrent Nets That Time and Count, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks 3, 189-194.