

乗車中の列車の各駅到着時間を自動表示する スマートフォンアプリケーション

井上 晴稀¹ 梶 克彦²

概要：列車を利用する場合，急いで列車に乗車して目的地までの詳細な到着時間がわからないといった事例が多い．車内で提示されている情報には目的地までの到着時間は提示していない．そのため本研究では，乗客のスマートフォン端末において自動的に乗っている列車を推定して，乗客に到着時間情報を提示するアプリケーションを開発する．本研究では，移動手段の一つのサービスと捉え，様々なサービスを提供する MaaS(Mobility as a Service) の考えを取り入れている．列車に乗っている乗客は，アプリケーションを開くだけで列車の情報と到着時間情報が何も操作せずに得られる．乗客のスマートフォン端末から取得した現在地情報を時系列化し，列車推定処理を行う Web サーバに送信する．その後，路線推定，進行方向推定，乗っている列車の推定を行う．推定した列車の情報から到着時間情報を抽出し，アプリケーションに送信して乗客に提示する．これまでは各駅停車において簡単な到着時間情報を提供するシステムはあったが，急行などの優等列車においては到着時間情報の提供ができなかった．本研究ではそのような優等列車においても到着時間情報の提示ができる処理を実装した．また，路線推定処理において複数路線が並列走行していると判断した場合には，乗客側に複数の路線候補を表示する．本研究で開発したアプリケーションにおいて，並列走行区間に対して隣接している路線が候補として表示されるか評価実験を行った．並列走行区間においては，隣接する路線も正しく表示できる区間と，そうではない区間の2つの結果を確認した．

Smartphone Application to Display The Arrival Time of A Riding Train

HARUKI INOUE¹ KATSUHIKO KAJI²

1. はじめに

日本では鉄道やバス・航空機といった様々な交通機関が発達しており，広く日常的に使われている．鉄道においては日本全国で広い路線網が発達している．バスにおいても地域内の路線バスや主要都市を結ぶ高速バスなどの路線が存在する．これらの移動手段は通勤や旅行など様々な用途で使用されている．

これらの移動手段の一つのサービスとして捉え，様々なサービスを提供する MaaS(Mobility as a Service) という考え方が近年注目されている．鉄道やバスなどの乗り物や運行している会社が異なっても，移動手段そのものに対して乗換経路検索や料金を支払うシステムをシームレス

に提供する考え方が MaaS である．この考え方によって乗り物を利用する人は，乗り物や目的地の情報など，移動手段に関する様々な情報を考えなくても移動を支えるサービスを利用できる．

乗り物を利用する乗客の中には，目的地まで行く列車やバスに急いで乗車する乗客がいる．同じ駅の中における乗り換えや，停留所が周辺にたくさんある場所では乗客もどれに乗っていいかわからない場合があり，とりあえず目的地まで行きそうな列車やバスなどに乗車する．急いで乗車した乗客は，自分自身が向かう目的地までの詳細な到着時間情報の情報を求め，車内に表示しているディスプレイや自分自身のスマートフォンにある経路案内アプリケーションにて自分の乗っている乗り物について情報を検索する．しかし，車内ディスプレイにおいては到着までの残り分数が直近の駅でしか表示されないだけでなく，広告など到着

¹ 愛知工業大学大学院 経営情報科学研究科

² 愛知工業大学 情報科学部情報科学科

時間情報とは関係のない情報が提示されている場合もある。また、経路案内アプリケーションにおいて乗り物の情報を表示させるために、利用者は発車した時間や発車した駅や停留所など、乗り物に関する情報を入力する必要がある。



図 1 本研究のアプリケーションの概要

そこで本研究では位置情報技術を使って、自動的に乗っている乗り物を推定し、様々な情報を表示するスマートフォンアプリケーションを開発を目指している。本研究のアプリケーションの概要を図 1 に示す。今までは自分が乗った乗り物に関する様々な情報を乗客が考慮する必要があった。しかし本研究では、乗客が様々な乗り物に関する情報を何も考慮せず、自分が持っているスマートフォンに乗っている乗り物の情報を表示する。乗り物の情報は、例えば乗っている乗り物の種別や行き先、到着時間情報や停車する場所における乗換案内などである。また、到着時間情報と列車情報は鉄道会社や路線、種別に関係なく表示できる。このように、本研究では乗客が鉄道会社や路線を考慮せずにサービスを利用できる MaaS の考え方を取り入れている。

本研究では、移動手段の中でも鉄道乗車中においてアプリケーションを起動しただけで列車の情報や到着時間情報を表示する。鉄道に焦点を絞った理由として、移動手段の中でも比較的正確な時間に運行されており、列車情報のオープンフォーマット化がしやすい点が挙げられる。乗客がアプリケーションを開くだけで乗客の乗っている列車の路線情報や列車の種別、行き先、停車する駅の到着時間情報が提示される。さらに他の路線と並列で走っている区間については、隣接している路線の候補を提示する。他の路線の候補を選択した場合、選択した路線の路線情報や列車の種別、行き先、停車駅の到着時間情報が改めて提示される。

本稿の構成は以下の通りである。第 2 章では関連研究を列挙する。第 3 章では列車の到着時間情報を自動表示するスマートフォンアプリケーションについて使用するデータや推定までのアプローチ、到着時間情報提示までの各処理の処理内容について説明する。第 4 章では本研究で開発し

たアプリケーションについての評価実験について述べる。第 5 章では本稿のまとめと今後の課題について述べる。

2. 関連研究

MaaS 関連では、日本におけるタクシーやカーシェアリングサービスの実情や公共交通機関の仕組みを考慮した MaaS モデルを構築し、構築したモデルを元にしたシステムアーキテクチャの検討を行っている研究 [1] が存在する。また、全ての移動手段に対して利用者が求める要件をバランスよく満たすルートを提案するアルゴリズムを構築している研究 [2] が存在する。この研究は、移動に関して利用者が求めるコストと待ち時間と快適さの 3 つの要求をバランスよく満たしてくれるルートを表示するアルゴリズムを構築し、検証実験を行っている。

様々な手法を用いた列車の位置を推定する研究は数多く行われている。スマートフォン端末の気圧センサを使って地下鉄の走行位置を推定する手法 [3] が提案されている。この手法は、予め事前に観測された気圧の変化や標高の変化等を記録しておき、記録した結果とスマートフォンで観測したデータと比較して走行位置推定を行っている。他にも磁気センサを使った地下鉄の発車および停車検知を行っている研究 [4] や、磁気センサを使って記録したデータを用いて地下鉄の列車の停車を検知し、停車した時間を地下鉄の運行スケジュールに反映するシステムの構築を行っている研究 [5]、スマートフォンに内蔵されている GPS を用いた到着時間予測システムの構築に関する研究 [6] が行われている。本研究で開発するアプリケーションにおいては、乗客の現在地情報は比較的容易に取得できる GPS を用いて取得するが、地下鉄区間においては GPS による現在地取得が難しい。そのため、スマートフォンのあらゆるセンサを用いて現在地情報を取得する手法も考慮しなければならない。

バスに対する到着時間予測に関する研究が、様々な手法で行われている。綿貫ら [7] は、実際の運行データを元に低ランク双線形ガンマ回帰モデルを使用して翌日以降の路線バスの所要時間推定を行っている。この他にも、人工ニューラルネットワークモデルを用いた手法 [8] や、k-近傍回帰アルゴリズムを用いた手法 [9]、カルマンフィルタを用いた手法 [10] で路線バスの所要時間推定を行っている研究が存在する。

鉄道の列車に対して到着時間を予測する研究も様々な手法が提案されている。Pongnumkul ら [11] は、タイの国営鉄道で提供されている列車遅延表示サービスを改善するため過去の旅行時間を用いた列車の到着時間予測を改善する手法を様々なアルゴリズムで構築している。過去の到着時間履歴を元に移動平均手法を用いて到着時間を予測するモデルと、同じく過去の到着時間履歴を元に k-近傍回帰手法を用いて到着時間を予測するモデルを実装し、列車の到

着時間を予測する際に発生した誤差を以前適用していたアルゴリズムに比べて縮小しているのを確認している。また、Liu ら [12] は鉄道輸送における逆伝搬ニューラルネットワーク、ウェーブレットニューラルネットワーク、遺伝的アルゴリズムを用いた到着時間予測モデルについて提案している。

本研究で開発するアプリケーションは列車情報の保存形式として GTFS を用いているが、運行情報等をリアルタイムで保存できる GTFS リアルタイムという規格が存在する。近年では、実際に GTFS リアルタイム規格を用いて様々なサービスを実装し、サービスに対する品質評価を行う研究も行われている [13][14]。

本研究で開発するアプリケーションに類似しているサービスとして、駅.locky[15] が挙げられる。このサービスはスマートフォンアプリケーションを起動しただけで、利用者の最寄り駅の発車時刻までの残り時間が提示されるサービスである。アプリケーションを起動しただけで情報が提示される点や、路線を気にすることなく最寄り駅の発車時刻までの情報が提示される点が、本研究で開発するアプリケーションと類似している。

3. 到着時間情報を自動的に提示するアプリケーション

乗客に対して列車の情報や到着時間情報を提示するためには、様々な情報が必要となる。例えば到着時間を表示するために必要な時刻表データや、種別や行き先といった列車に関する情報、軌道や駅などの地理的情報など様々な情報が必要となる。また、乗客のスマートフォン端末の位置情報システムを利用して取得した乗客の現在地情報も推定に必要な情報となる。

時刻表や列車の情報といったデータは、オープンフォーマット形式である GTFS(General Transit Feed Specification) 形式で保存する。GTFS とは、公共交通機関の列車や路線に関する情報や時刻表データ、軌道データ等の地理情報を格納できる Google 社が提供しているオープンフォーマット形式である。今まで列車に関する情報は、鉄道やバスの運行会社が独自で提供してきた独自フォーマット形式で保存されてきたが、GTFS を用いることによって鉄道やバスの情報を統一したフォーマットで保存でき、公共交通機関の運行会社や路線を考慮する必要なくデータを活用できる。また、アプリケーションを開発する人にとっても公共交通機関のデータを容易に取扱ができる。

鉄道路線の軌道や駅といった地理情報は、国土交通省の国土数値情報から GTFS や PostGIS を用いて緯度経度形式で保存する。国土交通省の国土数値情報では、鉄道の軌道を緯度経度形式で表現したデータと、駅の所在点を緯度経度形式で表現したデータが提供されている。軌道のデータは複数の緯度経度形式で表現された点で構成されている。

この軌道を表す複数の点を始点から終点まで並び替えをして、GTFS 形式に軌道情報として保存する。また、緯度経度形式で表現されている駅の地理データも GTFS 形式で保存している。また GTFS 形式とは別に、PostGIS に軌道の地理データを一つの線形データとして、駅の地理データを一つの緯度経度の点データとして保存している。PostGIS とは、PostgreSQL というデータベースシステムにおいて緯度経度で表現された地理情報を取扱ができる拡張機能である。PostGIS は地理情報を格納するだけでなく、2 点間の距離計算などの地理的演算もできるので、軌道の地理データや駅の地理データをデータベースに格納している。

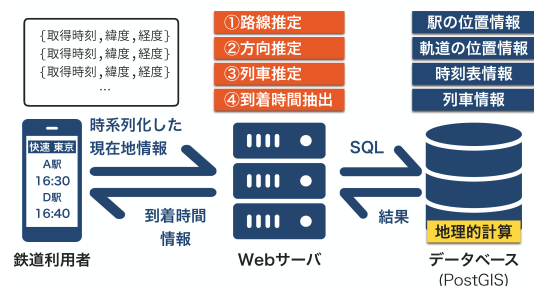


図2 アプリケーションのシステム概要

アプリケーションが自動的に乗っている列車を推定して到着時間情報を乗客に提示させるために、4つの処理を行う。システム概要を図2に示す。列車に乗っている乗客のスマートフォン端末から得られた現在地情報をWebサーバに送信する。ここで乗客の現在地情報をWebサーバに送信するとき、スマートフォン端末から取得した最新の現在地情報だけでなく、過去の所得した現在地情報に対しても、それぞれ時間要素を付加して時系列データとして変換した上で送信する。時系列化された現在地情報を取得したWebサーバは、次の4つの処理を行う。

- (1) 列車がどの路線を走行しているのか推定する処理
- (2) 列車がどの方向を進んでいるのか推定する処理
- (3) 実際に乗客が乗っている列車を推定する列車推定処理
- (4) 推定した列車情報から到着時間情報を抽出する処理

これらの4つの処理を行ったあと、乗客に対して乗っている列車の到着時間情報を提示する。Webサーバ内で4つの処理を行ったあと、列車の情報と到着時間情報を含んだデータをアプリケーションに対して送信する。アプリケーションは受け取ったデータから乗客に対して種別や行き先といった列車情報と次に停車する駅からの到着時間情報を提示する。

並列走行区間内においては推定した結果が正しくない可能性が高いため、隣接していると判断した区間においては隣接路線を候補として提示する。複数の路線が並走している区間では、スマートフォン端末を用いて取得した乗客の現在地情報に誤差が生じる可能性がある。誤差が生じた場

合、乗客に対して隣接している路線の到着時間情報を提示してしまう恐れがある。このような事象を防ぐため、並列区間と判定した場合は到着時間情報を提示する画面内において隣接路線の候補を同時に提示しておく。乗客が自分自身が乗っている路線を選択すると、選択した路線の到着時間情報を改めて表示する。Web サーバからアプリケーションへ送る到着時間情報も、路線推定処理の段階で並列走行区間と判定すれば路線数に応じた到着時間情報を送信する。

3.1 路線推定処理

乗客の列車を推定する前段階として、乗客の乗っている路線を推定する処理を行う。乗客の乗車している列車を推定するためには、まず乗客が乗車している路線の情報が必要となる。そのため Web サーバはアプリケーションから時系列化された現在地情報を受け取ったあと、一番最初に路線推定処理を行い、そのあと列車の方向推定や列車を推定する処理を行う。

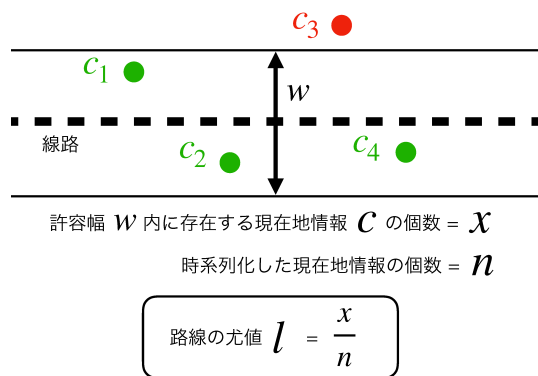


図 3 線路推定処理の機能概要

路線推定処理は、データベースに登録されている路線の軌道線形に対して許容幅を設定し、乗客の現在地情報が線形に対する許容幅に入っている確率を算出する。処理の概要を図 3 に示す。スマートフォン端末を用いて取得した時系列化された乗客の現在地情報を、取得された日時が古い順から軌道線形に対する許容幅の中に存在するのかが判定する。Web サーバは緯度経度形式で表現されている乗客の現在地情報と、データベースに格納されている軌道の地理情報に対して PostGIS を用いて、軌道の許容幅の中に存在しているかを判定する。

許容幅に入っているかを判定する処理を時系列化された乗客の現在地情報すべてで行ったあと、許容幅に入っている現在地情報の割合を算出する。図 3 に線路推定の概要を示す。軌道線形に対する許容幅を w とし、許容幅 w に入っている乗客の現在地情報 c の個数を x とする。このとき乗客の現在地情報の全体数を n とすると、許容幅に入っている現在地情報の割合 l は以下の式で計算できる。

$$l = \frac{x}{n}$$

このとき尤度 l は以下の数値の範囲内に収まる。

$$0.0 \leq l \leq 1.0$$

許容幅に入っている確率が最も高い路線を乗客が乗っている路線として推定する。乗客の現在地情報が線形に対する許容幅に入っている確率を算出する処理を、データベースに格納されている路線全てに対して行う。算出したすべての路線の確率の中から、最も確率の高い値が算出された路線を、乗客が乗車しているであろう路線として推定する。このとき、最も確率が高い値であるのにもかかわらず、確率の値が非常に低い場合または 0 の場合は、乗客が列車に乗っていないと判定する。

許容幅に入っている確率が最も高くない場合においても、並列走行路線と判定した場合には別の路線候補として路線情報を提示する。例えば、許容幅に入っている確率が 1.0 である路線が 2 つあった場合、2 つの路線は並列走行路線と判定する。並列走行路線と判定した場合、乗客がどの路線の列車に乗っているのかが判別が難しいため、1 つの路線を一旦乗客の乗っている路線と推定しつつ、もう一つの別の路線を路線候補として Web サーバ内あるいはアプリケーション内にて提示する。複数の路線が提示された場合、路線推定の後に行う進行方向推定や列車推定は、提示された複数の路線それぞれに対して処理を行う。

3.2 進行方向推定処理

路線推定処理を行ったと、Web サーバは乗客の乗っている列車がどの方向に進行しているのかが推定する処理を行う。乗客の乗っている列車を推定するためには、乗客が乗っている列車の進行方向の情報が必要である。なぜならば、列車の時刻表データは、始発駅から終着駅へ走行している列車と、終着駅から始発駅へ走行している列車の 2 つに分かれているためである。乗客の乗っている列車の進行方向を推定すると、列車の推定処理を容易にするだけでなく、乗客の乗っている列車の推定精度が上がる。

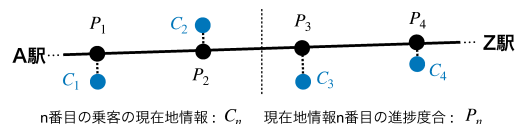


図 4 乗客の時系列現在地情報から算出する進捗度合の概要

Web サーバは、受け取った時系列化された現在地情報から乗客の乗っている列車の進行度合を算出し、時系列データとして保存する。乗客の端末から送られた時系列化された現在地情報から算出する進捗度合の概要を図 4 に示す。始発駅を A 駅、終着駅を Z 駅とした場合における時系列

データ n 番目の進捗度合を p_n で表す。進行度合 p は、0.0 が始発駅、1.0 を終着駅として、与えられた一つの緯度経度形式で表現された点が始発駅からどの程度進んでいるのかを表す割合である。この進捗度合は PostGIS において算出する。このとき、進捗度合 p は以下の範囲内になる。

$$0.0 \leq p \leq 1.0$$

乗客から送られた現在地情報の総個数を n としたときに、時系列化されている進捗度合 p を 2 分割し進捗度合 p の平均 μ を求める。時系列化した乗客の現在地情報を古い順から並び替えて、前半と後半に二分割する。時系列化した現在地情報の前半の個数 n_1 と、後半の個数 n_2 は以下の式で表せる。なお、計算上で発生した小数点は全て切り捨てる。

$$n_1 = \frac{n}{2} \quad n_2 = n - n_1$$

2 分割した時系列データそれぞれの平均 $\overline{p_1}, \overline{p_2}$ は次のように計算して求める。

$$\overline{p_1} = \frac{1}{n_1} \sum_{i=1}^{n_1} p_i$$

$$\overline{p_2} = \frac{1}{n_2} \sum_{i=n_1+1}^{n_2} p_i$$

平均値 $\overline{p_1}, \overline{p_2}$ を算出した後に、平均値の大小を比較して列車の進行方向を推定する。 $\overline{p_1} < \overline{p_2}$ の場合は Z 駅方向に、 $\overline{p_1} > \overline{p_2}$ の場合は A 駅方向に進行していると推定する。 $\overline{p_1} = \overline{p_2}$ の場合は方向が推定できない。そのため一旦処理を終了し、乗客のスマートフォン端末において推定処理中に取得した新しい現在地情報を含めて時系列化した現在地情報を再度 Web サーバに送信し、路線推定処理を行う。

3.3 列車推定処理



図 5 列車推定処理の概要

路線推定処理と進行方向推定処理が終わった後、乗客が乗っている列車の推定処理を行う。乗客の乗っている列車の推定に必要な路線の情報と列車の進行方向の情報は、列車推定処理の前に行った路線推定処理と列車進行方向推定処理において推定した。これらの推定した情報とアプリケーション側から受け取った時系列化された現在地情報から、乗客の乗っている列車を推定する。

急行などの優等列車においても到着時間情報が提示でき

るよう、対象路線の列車に対して推定走行位置を算出し、乗客の現在地情報との進行度合の差を時系列的に算出して列車推定を行う。処理の概要を図 5 に示す。乗客の時系列化された現在地情報に対して取得日時における進行度合を算出し、対象路線の全列車の推定進行度合との差を算出する。進行度合の差を時系列データとして保存し、二乗平均平方根を用いて乗客が乗っている列車を推定する。対象路線の全列車の進行度合を求める手法により、各駅停車だけでなく優等列車に対しても列車推定を行える。

最初に、Web サーバは処理を開始する段階の日時で走行している列車を抽出する。線路推定において推定した路線情報と、方向推定処理で算出した方向情報を用いて、データベースから該当する路線および方向の列車情報を抽出する。抽出した時刻表情報から始発の時間と終着の時間を取り出して、列車が推定処理を開始する日時において走行しているか判定する。処理を開始する日時において走行していると判断した列車情報を、アプリケーション内に保持する。

次に、時系列化された乗客の現在地情報に対して、抽出した全ての列車の推定走行位置を算出する。ある一つの列車情報に注目し、時系列化された乗客の現在地情報に記録されている日時情報から、計算対象である列車の直前に発車した駅の出発時刻と、到着する駅の到着時間を算出する。到着する駅の到着時間と直前に発車した駅の出発時間との差を秒単位で求め t_1 とする。また直前に発車した駅と次に到着する駅の距離を d_1 とする。これらより、列車の表定速度 m を以下の式から算出する。

$$m = \frac{t_1}{d_1}$$

次に、時系列化されている現在地情報に記録されている時刻と直前に発車した駅の出発時刻との差を秒単位で求め t_2 とする。 t_2 と表定速度 m を使い、以下の式から推定進行距離 d を算出する。

$$d = t_2 m$$

算出した推定走行距離 d から、軌道に対する列車の推定進行度合を求め、時系列化された現在地情報の時刻情報から算出した軌道に対する乗客の進行度合との差を計算する。この計算を、全ての時系列化された現在地情報に対して繰り返し行う。その後、進行度合の差に対して二乗平均平方根を算出する。二乗平均平方根は、算出した進行度合の差に平方した値を全て足し合わせ、現在地情報の時系列データの個数分に対する平均値を算出し、算出した値の平方根を取る。ここで、時系列化された現在地情報の時刻情報から算出した乗客の進行度合と軌道に対する列車の推定進行度合の差を b 、時系列データの個数を n とした場合、行度合の差に対する二乗平均平方根 $RMS[b]$ は以下の式で算出

する。

$$RMS[b] = \sqrt{\frac{1}{n} \sum_{i=1}^n (b_i)^2}$$

二乗平均平方根 $RMS[b]$ を、処理を開始する日時において走行していると判断した全ての列車に対して算出し、 $RMS[b]$ の値が一番低い列車を、乗客が乗車している列車として推定する。乗客が乗車していると推定した列車の情報を、アプリケーション内で保持する。

3.4 到着時間情報の抽出および提示

急行	名鉄一宮
男川	20:57
東岡崎	21:00
新安城	21:07
知立	21:12
前後	21:17
鳴海	21:22
堀田	21:28
神宮前	21:33
金山	21:37

名古屋本線

図 6 本研究で開発するアプリケーションの表示例

推定した列車情報を元に、時刻表データから推定した列車情報の到着時間情報と列車の旅行程情報を抽出する。乗客の現在地情報から一番近い列車と推定した情報を元に、時刻表データベースから当該列車の時刻表を抽出する。時刻表データベースの中には停車駅と出発時刻、発車時刻の情報が主に記述されており、その中から停車駅と到着時刻の情報を選択し抽出する。また、当該列車の情報から種別や行先といった旅程に関する情報もデータベースから抽出する。抽出した旅程情報と停車駅情報、到着時間情報を JSON 形式に変換しアプリケーションに JSON 結果を返す。

到着時間情報を JSON 形式で受け取ったアプリケーションは、推定した列車の到着時間情報を乗客に提示する。アプリケーションにおける提示画面を図 6 に示す。アプリケーション内では種別や行先といった情報と、到着時間情報が提示される。並列走行区間内と判定された場合は、隣接する路線がアプリケーション下部に候補として列挙され、乗客が路線名を選択する。その後、選択した路線名の到着時間情報に切り替わる。

4. 評価実験

本研究で開発したアプリケーションが、並列路線区間内で正しく複数路線候補が提示されるか評価実験を行った。路線推定において複数路線と推定した場合には、アプリケーションの下部に隣接して走行する路線を乗客に対して提示する。しかし、乗客がアプリケーションを起動した段階において推定された路線が単独走行区間であり、途中で異なる路線が合流した場合は隣接する路線を提示する必要がある。また、アプリケーション起動段階では並列走行区間内であったものの、途中で路線が別々に分かれるときは、乗客の乗っている路線に提示を移行する必要がある。並列区間内においての複数路線提示においては、路線が分岐した場合、または合流する場合を考慮して評価実験を行う。

この評価実験においては、地下空間での使用および乗換に関しては考慮していない。本研究で開発しているアプリケーションは GPS を使用している。そのため、地下空間内で現在地情報を取得するのは難しい。そのため、実際のスマートフォン端末で現在地情報を取得し、それを用いてのシミュレーションを行う場合は地下走行区間を除外している。また、アプリケーションにおいては並列走行区間内において別の路線に乗換した後に、乗車している路線および列車の再推定はしない。そのため、並列走行区間の乗換に関しても評価実験内において考慮しない。

4.1 路線推定に用いる許容幅 w の設定



図 7 許容幅 w を 140m に設定した場合

路線推定に用いる許容幅 w は、実際に列車に乗車してスマートフォン端末を用いて取得した時系列化された現在地情報を用いて設定する。今回は、名古屋鉄道の名古屋本線に実際に乗車し、実際にスマートフォン端末を用いて GPS 技術を使用して取得した現在地情報を時系列化して保存し、地図上に路線の軌道と現在地情報を描画した上で許容幅 w を設定する。GPS を使用した現在地情報の取得は、Xperia XZ1 を使用して取得した。以後、現在地情報の取得で用いるスマートフォン端末は同じ端末を用いる。

乗客の乗っている路線の推定処理において図 3 内の許容



図 8 駅停車中における現在地情報のズレ

幅 w を 140m に設定して路線推定処理を行う。図 7 において、許容幅 w を 140m に設定した場合における並列走行区間の軌道に対する許容幅を地図上に表した図を示す。図に示すとおり、名鉄の名古屋本線の軌道を中心として許容幅 w を 140m に設定した場合、十分に隣接路線である JR 東海の東海道線の軌道を覆っているのが確認できる。これにより、隣接する路線があった場合は複数路線候補として乗客に提示できる。また、駅停車中における乗客の現在地情報を図 8 に示す。駅に停車した場合は、GPS を用いて取得した現在地情報がデータベースに登録されている軌道よりズレが生じているのが確認できる。図 8 のように許容幅 w を 140m に設定すると、駅停車中にズレが生じた場合においても許容幅内に収まると確認した。こうした理由から、路線推定における許容幅 w を 140m に設定した。

4.2 複数路線候補の提示基準値の設定

次に、並列走行区間内から単独走行区間へと変化した場合や、起動時に単独走行区間だったが後に並列走行区間に変化した場合において、複数路線候補として隣接路線を提示候補から外すための基準値を設定する。実際にスマートフォン端末から取得した時系列化された現在地情報を用いて路線推定シミュレーションを行い、算出された各路線の尤度をグラフを用いて可視化して候補から外すための基準値を設定する。今回は並列走行区間として、名古屋鉄道の名古屋本線にある堀田駅から神宮前駅間における時系列化された現在地情報から各路線の尤度をグラフで表す。名鉄



図 9 名鉄名古屋本線 堀田から神宮前間の合流・分岐地点

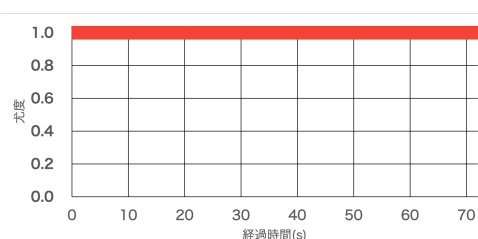


図 10 名鉄の路線推定の尤度の推移 (堀田から神宮前)

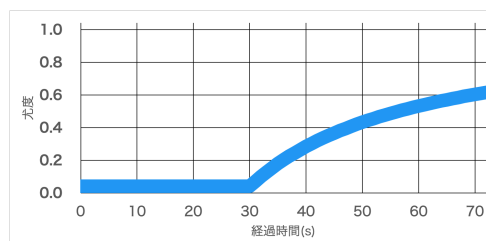


図 11 JR 東海の路線推定の尤度の推移 (堀田から神宮前)

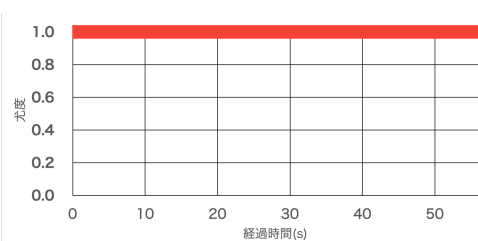


図 12 名鉄の路線推定の尤度の推移 (神宮前から堀田)

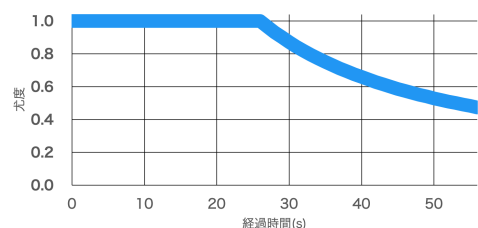


図 13 JR 東海の路線推定の尤度の推移 (神宮前から堀田)

名古屋本線の堀田駅から神宮前駅の地図を図 9 に示す。名鉄名古屋本線の堀田駅から神宮前駅間は、途中で JR 東海の東海道線と合流または分岐する区間である。この区間内で異なる路線が合流する場合における尤度の推移グラフを図 10 と図 11 に、分岐する場合における尤度の推移グラフを図 12 と図 13 に示す。

路線推定の尤度の推移グラフから、異なる路線が合流または分岐した場合において乗客に対して提示候補から外すための基準値を設定する。尤度が最も高かった路線の尤度を a_1 、任意の路線の尤度を a と置いたとき、以下の式に当てはまる任意の路線は他の路線の候補として乗客に路線候補を提示、到着時間情報を提示する。

$$a_1 - a < 0.1$$

4.3 複数路線候補の提示



図 14 名鉄名古屋本線 豊橋から伊奈

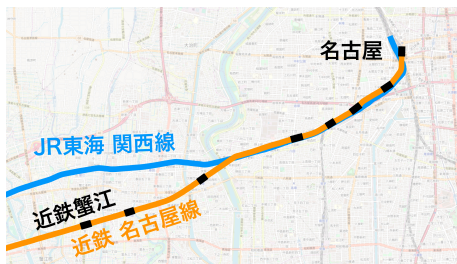


図 15 近鉄名古屋線 名古屋から蟹江

4.1 節と 4.2 節から許容幅と複数路線候補の基準値を設定した上で、並列走行区間内において乗客に対して複数路線候補が正しく提示されるか実験を行った。実際に並列走行区間内において列車に乗車してスマートフォン端末から取得した現在地情報を時系列化して保存し、それを用いてアプリケーションの動作をシミュレーションして提示の確

認を行う。評価実験を行う並列走行区間を以下に示す。また、図 9 に示している堀田から神宮前の地図の他に、名鉄名古屋本線の豊橋から伊奈までの地図を図 14 に、近鉄名古屋線の近鉄名古屋から近鉄蟹江の地図を図 15 に示す。

- 名鉄名古屋本線 豊橋～伊奈（JR 東海道線と隣接）
- 名鉄名古屋本線 堀田～神宮前（同上）
- 近鉄名古屋線 近鉄名古屋～近鉄蟹江（JR 関西線と隣接）

特急	名鉄岐阜
国府	8:52
東岡崎	9:06
新安城	9:12
知立	9:16
神宮前	9:30
金山	9:33
名鉄名古屋	9:38
国府宮	9:50
名鉄一宮	9:55

分岐後

特急	名鉄岐阜
国府	8:52
東岡崎	9:06
新安城	9:12
知立	9:16
神宮前	9:30
金山	9:33
名鉄名古屋	9:38
国府宮	9:50
名鉄一宮	9:55

図 16 隣接路線の分岐におけるアプリケーションの動作

特急	名鉄岐阜
国府	8:52
東岡崎	9:06
新安城	9:12
知立	9:16
神宮前	9:30
金山	9:33
名鉄名古屋	9:38
国府宮	9:50
名鉄一宮	9:55

合流後

特急	名鉄岐阜
国府	8:52
東岡崎	9:06
新安城	9:12
知立	9:16
神宮前	9:30
金山	9:33
名鉄名古屋	9:38
国府宮	9:50
名鉄一宮	9:55

図 17 隣接路線の合流におけるアプリケーションの動作

異なる路線のみ提示

快速	亀山(三重県)
桑名	9:27
四日市	9:39
南四日市	9:43
河原田	9:47
河曲	9:52
加佐登	9:56
井田川	10:00
亀山	10:06

乗車中の路線のみ提示

急行	松阪
近鉄蟹江	9:29
近鉄物産	9:35
桑名	9:43
近鉄富田	9:50
近鉄四日市	9:56
塩浜	10:00
伊勢若松	10:08
白子	10:13
江戸橋	10:22

図 18 近鉄名古屋線 名古屋から近鉄蟹江における動作

名鉄名古屋本線の豊橋から伊奈におけるアプリケーションの動作を図 16 と図 17 に示す。アプリケーション起動時に並列走行区間を走行していた際は、隣接する路線が複数路線候補として提示されている。その後、隣接している異なる路線が分岐してしばらく経過すると、乗客が乗って

いる名鉄名古屋本線の情報だけを提示すると確認した。また、単独走行区間からアプリケーションを起動した場合は、乗客の乗っている名鉄名古屋本線だけの情報を提示していた。その後、異なる路線が合流した場合においても、乗客の乗っている名鉄名古屋本線だけの情報を提示され続けているのを確認した。名鉄名古屋本線の堀田から神宮前においても同様の動作を確認した。

近鉄名古屋線の近鉄名古屋から近鉄蟹江におけるアプリケーションの動作を図 18 に示す。この区間においては、並列走行区間内にアプリケーションを起動した場合に、異なる路線が単独提示されてしまうのを確認した。また、並列路線が分岐した後は乗客の乗っている近鉄名古屋線の情報が単独表示になるのを確認した。並列区間内においては近鉄の路線と JR の路線ともに路線推定の尤度が 1.0 を示していた。よって、路線推定処理内ではなく他の処理における不具合であると考えた。

5. まとめ

本研究では乗客の乗車している列車を推定し、到着時間情報を表示するアプリケーションを開発した。移動手段の一つのサービスとして提供する MaaS という考えを取り入れたサービスが数多く登場している。本研究においても、鉄道の乗客がスマートフォンアプリを開くだけで鉄道の運行会社を気にする必要なく、乗客の乗っている列車の到着時間情報を乗客に対して提示する。並列走行区間内を走行中と推定した場合、乗客に対し路線候補として隣接している路線を提示する。

並列走行区間内において隣接路線が正しく候補として提示されるか評価実験を行った。並列走行区間内においては、正しく隣接路線を提示した区間はあったものの、異なる路線だけが提示される区間も存在した。線路推定における尤度は正しく算出されているため、路線推定以外の処理における不具合と考えた。

今後の課題は、現状の路線推定における不具合の修正、地下走行区間における列車推定の実装が挙げられる。評価実験から、路線推定あるいは列車推定に何らかの不具合が発生しているため、修正する必要がある。また、地下走行区間の列車推定については GPS だけでなく、スマートフォン端末に搭載しているセンサを用いて実装を図っていく。

参考文献

- [1] 日高洋祐: 日本版 MaaS (Mobility as a Service) モデルのシステムアーキテクチャの検討, 研究報告高度交通システムとスマートコミュニティ (ITS), Vol. 2018, No. 10, pp. 1–6 (2018).
- [2] Kamau, J., Ahmed, A., Rebeiro-H, A., Kitaoka, H., Okajima, H. and Ripon, Z. H.: Demand responsive mobility as a service, *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, pp. 1741–1746 (2016).
- [3] Hyuga, S., Ito, M., Iwai, M. and Sezaki, K.: An on-line localization method for a subway train utilizing the barometer on a smartphone, *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, p. 50 (2016).
- [4] Shin, H., Lee, G. and Han, D.: Subway Stop/Departure Detection using a Magnetic Sensor of the Smartphone, *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, ACM, p. 37 (2018).
- [5] Lee, G. and Han, D.: Subway train stop detection using magnetometer sensing data, *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 766–769 (2014).
- [6] Jayawardena, A. N., Sachith, K. and Wijekoon, J. L.: Trainko: Poster Abstract: Train Arrival Time Prediction System for Sri Lanka, *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '18, Piscataway, NJ, USA, IEEE Press, pp. 110–111 (2018).
- [7] 綿貫圭太, 下坂正倫: バス運行状況ウェブサービス情報を用いた乗換案内アプリのための路線バス所要時間推定, 研究報告ユビキタスコンピューティングシステム (UBI), Vol. 2018, No. 5, pp. 1–8 (2018).
- [8] Jeong, R. and Rilett, R.: Bus arrival time prediction using artificial neural network model, *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*, pp. 988–993 (2004).
- [9] Chang, H., Park, D., Lee, S., Lee, H. and Baek, S.: Dynamic multi-interval bus travel time prediction using bus transit data, *Transportmetrica*, Vol. 6, No. 1, pp. 19–38 (2010).
- [10] Bai, C., Peng, Z.-R., Lu, Q.-C. and Sun, J.: Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes, *Computational intelligence and neuroscience*, Vol. 2015, p. 432389 (2015).
- [11] Pongnumkul, S., Pechprasarn, T., Kunaseth, N. and Chaipah, K.: Improving arrival time prediction of Thailand's passenger trains using historical travel times, *2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 307–312 (2014).
- [12] Liu, Y., Tang, T. and Xun, J.: Prediction algorithms for train arrival time in urban rail transit, *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6 (2017).
- [13] Barbeau, S. J.: Quality Control-Lessons Learned from the Deployment and Evaluation of GTFS-realtime Feeds, *Transportation Research Board 97th Annual Meeting*, pp. 7–11 (2018).
- [14] Steiner, D., Hochmair, H. and Paulus, G.: Quality assessment of open realtime data for public transportation in the netherlands, *GI-Forum J. Geogr. Inf. Sci.*, Vol. 2015, pp. 579–588 (2015).
- [15] 矢野幹樹, 岩崎陽平, 河口信夫: 駅.Locky: 無線 LAN 位置推定を用いた時刻表アプリの開発, 全国大会講演論文集, Vol. 72, pp. 289–290 (2010).