Bachelorarbeit

Comparison of supervised learning algorithms for arrival time estimation in a dynamic vehicle routing setting

an der Technischen Universität
Carolo-Wilhelmina zu Braunschweig
Carl-Friedrich-Gauß-Fakultät
Institut für Wirtschaftsinformatik, Abteilung Decision Support.

Eingereicht von:	Emre Gezer
Matrikelnummer:	4901507
Studiengang:	Wirtschaftsinformatik
Referent:	Jun.-Prof. Dr. Marlin Ulmer
Betreuer:	M. Sc. Florentin D. Hildebrandt
Eingereicht:	Braunschweig, den 10.02.2021

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Braunschweig, 10.02.2021

Contents

List of Tables

List of Figures

List of Abbreviations

ML	Machine Learning
ETA	Estimation of arrival time

1 Introduction

Nobody likes waiting - be it when you order your next book on Amazon for class or the pizzas at UberEATS for your birthday. In fact, when doing research on this topic, psychologists found out that increased waiting times generally have a significant negative impact on customer satisfaction and loyalty [1,2]. If we take a glance at the online food delivery market, there are more than 700 million people globally that used food delivery services in 2017 with twice as many users being expected in 2024 [3]. According to forecasts for the same time frame, the userbase of eCommerce platforms in general will grow from 2.480 billion to roughly 4.6 billion users [4]. Thus, waiting becomes a large scaled economic issue. What one might not expect is that the customers' own perceptions regarding their waiting time negatively affect the perceived service quality stronger than actual waiting times do [5,6]. Given these points, we can conclude that a key challenge lies in the communication of accurate arrival time estimations to the customers.

Current research tackles this task commonly by means of machine learning techniques. Exemplary, while Hildebrandt and Ulmer used Gradient Boosting Decision Trees in their offline approach to map state features directly to expected arrival times in food delivery [7], Zhu et al. implemented a Multilayer Perceptron to predict accurate Order Fulfillment Cycle Times [8]. Hardly any kind of machine learning model comparison for arrival time estimation purposes in vehicle routing settings can be found. Therefore, we intend to conduct further research in this area.

This paper examines the forecast quality and performance of different supervised machine learning (ML) techniques in the context of arrival time estimation problems (ETA). To accomplish this task, the algorithms will be trained on historical meal delivery data collected within Iowa City.

This paper is organized as follows. Firstly, we will review and discuss literature that focuses on arrival time estimation via supervised learning. We will then proceed with our model, where we describe the ETA problem generally and define the model with respect to the underlying problem of our experimental design. Several steps are taken in the solution approach that follows upon our model definition. In the beginning, we give an introduction to fundamental concepts in supervised learning and present techniques used at different stages in the machine learning pipeline generally for the sake of understanding the way each of them functions. We will then advance to the computational study, where we present our experimental design and discuss the results. Finally, we will create variations of our experimental design in order to analyze each algorithm's robustness and impact in these different settings.

2 Literature Review

This chapter gives an overview of related research. [Ziel noch formulieren]

2.1 Most related work

This section includes research on offline arrival time estimation via supervised learning in dynamic pick-up and delivery settings. To the best of the authors knowledge, there are only three papers that fit this description. Amongst them, the most closely related work to this paper is that of Hildebrandt and Ulmer (2020), who contributed a offline supervised learning approach to predict arrival times for the Restaurant Meal Delivery problem, a dynamic pick up and delivery problem with uncertainty in travel times, processing times and requests originally presented in Ulmer, Thomas, Campbell, and Woyak (2020). In their offline approach, Hildebrandt and Ulmer (2020) map spatial, temporal, routing, and processing features based on the RMDPEAT to expected arrival times by means of a gradient boosting decision tree (GBDT) model. This paper is inspired by them and can be seen as complementary to their paper since we aim to estimate arrival times offline based on the same underlying problem setting via several supervised learning algorithms, including GBDTs.

Zhu et al. (2020) predict arrival times by means of deep learning with uncertainty being present in requests, courier travel times, courier waiting times at restaurants and cooking times. Besides using temporal, spatial and processing features for travel time prediction, they additionally include dish specific features and information about weather conditions. In contrast to Hildebrandt and Ulmer (2020), they include no routing information. They instead introduce a separate component that ranks courier assignments w.r.t. logistics cost and customer inconvenience. According to their analysis, the proposed deep learning architecture produces, inter alia, more accurate results than a GBDT approach.

Liu, He, and Max Shen (2018) compare linear models, support vector regression and ensemble learning methods for travel time estimation based on spatial, temporal and order-related features, and integrate travel time prediction into the order assignment problem with uncertainty in requests, travel times and service times. The order assignment problem aims to assign orders in a way that the assignments minimize the total delivery delay over all driver routes. Analysing the prediction models w.r.t to their accuracy, tractability and interpretability, they found that random forests (RF) and support vector regression yield slightly more accurate results but are computationally less tractable due to exponential runtime and less interpretable than linear models. For the latter two reasons, Liu et al. (2018) prefer linear models. Amongst the linear models, lasso regression obtained the best accuracy.

2.2 Arrival Time Estimation

In this section, we broaden the scope from offline arrival time estimations via supervised learning for dynamic pick-up and delivery problems to offline arrival time estimations via supervised learning for vehicle routes in general. Tab. 1 classifies the literature on arrival time estimation for vehicle routes with regards to the problem setting and the solution. With *Route type*, the table distinguishes arrival time estimation research that has been applied to vehicle route sequences consisting of single origin-destination (OD) pairs (referred to as *OD* in the table) from those that have been applied to route sequences consisting of multiple OD pairs (referred to as *trips* in the table) each. By *Uncertainty*, the table refers to uncertain elements in the underlying problem settings. Sources of Uncertainty considered here are requests, travel and service times, and processing times. Uncertainty in requests indicates that customer requests are not certainly known at the start of the problem and arrive dynamically over time. Uncertainty in travel and service times expresses itself through uncertain weather conditions, traffic congestion or individual challenges when serving customers (e.g. parking or waiting times). Uncertainty in processing occurs when two stochastic processes are synchronized (e.g. the synchronization of bus ride and bus boarding, or meal preparation and delivery). From the solution view, we classify the literature based on features and supervised learning algorithms used for travel time prediction. The *features* column distinguishes between temporal, spatial, routing and processing features. Temporal and spatial features include time and space related variables respectively. Examples for former are time stamps at a start/end point or historical travel times, and examples for latter are GPS coordinates or distances between locations. Processing and routing features give information about the uncertainty in processing times (e.g. meal preparation or bus boarding time) and the properties of a trip (e.g. number of stops in a trip) respectively. The column *Offline Approaches* presents all offline supervised learning approaches used in the respective literature, regardless of whether they were used as the primary approach to predict arrival times or solely for comparison purposes.

A noticeable amount of research on arrival time prediction for vehicle trips via supervised learning has been done in the field of bus arrival time prediction. Chen, Liu, Xia, and Chien (2004) estimate arrival times for bus trips based on manually selected spatial, temporal, and routing information via neural networks. [Compare with related papers that estimate arrival times for bus trips].

A significant amount of arrival time estimation research has also been done for origin-destination problems in different subfields of intelligent transportation systems.

To predict travel times on freeways for different short-term forecasting horizons, Vanajakshi and Rilett (2007) use support vector regression (SVR) based on estimated route travel times from prior research and conclude that SVR performs comparably well to artificial neural networks (ANN). Siripanpornchana, Panichpapiboon, and Chaovalit (2016) propose a deep learning architecture consisting of a deep belief network and a sigmoid regression layer. Former learns features in an unsupervised fashion based on historical route travel times as inputs, latter then estimates travel times based on these learned features. Cheng, Li, and Chen (2019) make use of GBDTs using manually selected travel time features and traffic state related variables. They report that the ensemble learning approach with GBDTs outperforms feedforward neural networks and support vector machines.

For taxi travel time prediction, Jindal et al. (2017) propose a unified approach based on raw NYC taxi data. They concatenate two neural networks, where the first one uses spatial features to predict travel distances, and the second one uses these predicted distances and additional temporal information to predict travel times. They solely compared their approach to other deep learning architectures. In contrast to them, Huang and Xu (2018) and Huang, Pouls, Meyer, and Pauly (2020) compare several tree-based learning methods to predict travel times on different horizons each based on NYC taxi data as well, among them random forests (both), GBDTs (both), and CART (only Huang et al. (2020)). While Huang and Xu (2018) selected features by means of principal component analysis, Huang et al. (2020) engineered them manually. Both ended up using spatial and temporal features mainly. Their results indicate that all tree-based ensemble methods are able to predict travel times more accurately than the respective benchmark algorithms (CART and naive approach in Huang et al. (2020); linear and logistic regression in Huang and Xu (2018)).

2.3 Summary

3 Problem statement

4 Methodology

This chapter presents the solution approach. Section 4.1 motivates our problem. Section 4.2 discusses the selected features. Section 4.3 explains the algorithms considered in the comparison in detail. Section 4.4 defines the process by which we evaluate the algorithms in order to assess the quality of each algorithm.

4.1 Algorithms

This section gives a short introduction to the conceptual framework of machine learning and then examines the algorithms included in our comparison. For further research, the interested reader is referred to Bishop (2006) and Wolf (2020).

Supervised learning algorithms receive a set of pairs $\{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_3, \mathbf{y}_3)\}$ where $x_i \in X^{N \times P}$ is a vector representing the i-th observation described by P features, and $y_i \in Y^{N \times 1}$ being the target variable that has to predicted.

Predictions can generally be made for either a **regression** or **classification** problem, which differ in the nature of their target variables. While regression is used to predict continuous targets, classification finds its use in the prediction of discrete targets. The arrival time estimation problem is a regression task since we are estimating (continuous) arrival times. Although various supervised learning models use different mathematical procedures to predict targets, all of them follow the principle of induction, meaning that general rules are inductively inferred from the observations. This is also called **generalization**. In order for a model to generalize the data properly, two things have to be avoided: **Underfitting** and **Overfitting**. Indicators for a underfitted model are poor predictions on both the training and test data, whereas a overfitted model does very well on the training data, but poorly on test data. Technically speaking, they occur when a model has either too few or many free available respectively to describe the underlying mapping of given training data.

4.1.1 Linear Models

Linear regression models assume a linear relationship between dependent and independent variables, i.e. inputs and outputs. They aim to fit a linear equation of the form

$$\hat{Y}(\mathbf{w}, \mathbf{X}) = b + \sum_{i=1}^{N} w_1 x_1 + w_2 x_2 + \cdots + w_p x_p \tag{4.1}$$

to observed data, where $b \in \mathbb{R}$ is the intercept, $\{w_1, \ldots, w_p\}$ are the weights and $\mathbf{X}$ is a matrix of dimensionality $p \times N$, i.e a matrix consisting of p features and N samples. Matrix notation is usually used for simplification purposes:

$$\hat{Y}(\mathbf{w}, \mathbf{X}) = b + \mathbf{w}^{\mathbf{T}}\mathbf{X} \tag{4.2}$$

4.1.2 Ensemble Learning

Ensemble learning techniques build a prediction models based on so-called *weak learners.* In machine learning, weak learners -Bagging

-Boosting

The first step in GBDT is to initialize a model by taking the derivate of the loss function with a known target y_i and an unknown target prediction γ that is constant for all corresponding y_i.

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{4.3}$$

Given the initial model $F_0(x)$ respectively the former learner $F_{m-1}(x)$, M regression trees can now be build sequentially as shown in the following. The first step computes pseudo-residuals r_{im} between the prediction value of the former learner for the i-th observation $F_{m-1}(x)$ and the actual i-th target for the m-th tree $y_i$by deriving the loss function w.r.t to $F_{m-1}(x)$:

$$r_{im} \;\; = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{4.4}$$

for every i-th example.

In the second step, the algorithm fits a regression tree to every r_{im} and creates disjoint regions (in decision tree terminology $\widehat{=}$ „leaves“) R_{jm} for $j = 1, ..., J_m$.

In the third step, we compute:

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{4.5}$$

for every terminal region j in the m-th tree. Given this, we now update the former weak learner with the new one:

$$F_m(x) = F_{m-1}(x) + \alpha \sum_{j=1}^{J_m} \gamma_m I(x \in R_{jm}) \tag{4.6}$$

where α is the learning rate.

4.1.3 Neural Networks

4.2 Feature Selection

As shown, researchers used different feature selection methods. A majority of the presented papers crafted their features manually relying on their domain expertise, whereas others (e.g Siripanpornchana et al. (2016) and Huang and Xu (2018)) used representation learning techniques.

- Raw Data - Manual feature selection - Feature learning via deep autoencoder

4.3 Evaluation

- Use MSE, MAE, MAPE etc. to derive infos about accuracy - Hyperparameter sensitivity analysis (Random Search) for robustness - Variations in data set for robustness Noise einführen –> Wie reagiert Algorithmus auf Noise Anzahl der Trainingsdaten –> Wieviele Samples bis Konvergenz? Feature Selection –> Welches Set optimiert bzgl. Runtime und Accuracy Feature Complexity –> Was passiert, wenn der Datensatz komplexer wird? (Veränderungen von Elementen aus Problemstellung wie z.B. Autos, Kunden, Restaurants)

Runtime

5 Computational Study

5.1 Experimental Design

5.2 Parametrization of Methods

6 Analysis

- Variations in experimental design
- Introduce noise to data - Change model assumptions (Changes in fleet, customer, restaurant variables, general assumptions (?)) - Hyperparameter sensitivity - Discuss Interpretability

7 Conclusion

References

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics).* Berlin, Heidelberg: Springer-Verlag.
6

Chen, M., Liu, X., Xia, J., & Chien, S. I. (2004). A dynamic bus-arrival time prediction model based on apc data. *Computer-Aided Civil and Infrastructure Engineering*, *19*(5), 364-376. Retrieved from `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.2004.00363.x` doi: https://doi.org/10.1111/j.1467-8667.2004.00363.x
3

Cheng, J., Li, G., & Chen, X. (2019). Research on travel time prediction model of freeway based on gradient boosting decision tree. *IEEE Access*, *7*, 7466-7480. doi: 10.1109/ACCESS.2018.2886549
4

Hildebrandt, F., & Ulmer, M. (2020, 08). *Supervised learning for arrival time estimations in restaurant meal delivery.*
2

Huang, H., Pouls, M., Meyer, A., & Pauly, M. (2020). *Travel time prediction using tree-based ensembles.*
4

Huang, L., & Xu, L. (2018). Research on taxi travel time prediction based on gbdt machine learning method. In *2018 eighth international conference on instrumentation measurement, computer, communication and control (imccc)* (p. 718-722). doi: 10.1109/IMCCC.2018.00155
4, 7

Jindal, I., Tony, Qin, Chen, X., Nokleby, M., & Ye, J. (2017). *A unified neural network approach for estimating travel time and distance for a taxi trip.*
4

Liu, S., He, L., & Max Shen, Z.-J. (2018, 05 15). On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science.* Retrieved from `https://doi.org/10.1287/mnsc.2020.3741` doi: 10.1287/mnsc.2020.3741
2

Siripanpornchana, C., Panichpapiboon, S., & Chaovalit, P. (2016). Travel-time prediction with deep learning. In *2016 ieee region 10 conference (tencon)* (p. 1859-1862). doi: 10.1109/TENCON.2016.7848343
3, 7

Ulmer, M., Thomas, B., Campbell, A., & Woyak, N. (2020, 08). The restaurant meal delivery problem: Dynamic pick-up and delivery with deadlines and random

ready times. *Transportation Science*. doi: 10.1287/trsc.2020.1000
2

Vanajakshi, L., & Rilett, L. R. (2007). Support vector machine technique for the short term prediction of travel time. In *2007 ieee intelligent vehicles symposium* (p. 600-605). doi: 10.1109/IVS.2007.4290181
3

Wolf, M. M. (2020). *Mathematical foundations of supervised learning.* Retrieved from `https://www-m5.ma.tum.de/foswiki/pub/M5/Allgemeines/MA4801_2020S/ML_notes_main.pdf`
6

Zhu, L., Yu, W., Zhou, K., Wang, X., Feng, W., Wang, P., ... Lee, P. (2020). Order fulfillment cycle time estimation for on-demand food delivery. New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi.org/10.1145/3394486.3403307`
2