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 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], whereas Ulmer and Thomas used approximation to estimate mean arrival times for values of aggregated states in service routing [9]. Other than Khiari et al. who compared several boosting and linear regression algorithms [10] (without incorporating neural networks in their comparison and spatial features in their underlying model which is what we will do), 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 on machine learning in general and its usage in the field of arrival time estimation. 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 present the used ML techniques and define them generally for the sake of understanding the way each of them functions. Concretely, our dataset will undergo a transformation process with the goal in mind of having a reduced feature space with the most impactful features left in the end. We will then advance to the computational study, where we present its experimental design and discuss its 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 presents prior research on offline arrival time estimation via supervised learning. The goal is to give the reader an overview about the

2.1 Most related work

This work is mostly inspired by Hildebrandt and Ulmer (2020), who contributed a supervised offline and online-offline approach to predict arrival times in the Restaurant Meal Delivery problem setting, a dynamic pick up and delivery problem with uncertainty in travel times, processing times and order bundling 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 can be seen as complementary to their paper since we aim to estimate arrival times based on the same underlying problem setting via several supervised learning algorithms, including GBDTs. Due to hardware limitations, partial and full online approaches are excluded from this study.

Other than Hildebrandt and Ulmer (2020), research on arrival time estimation via supervised learning in dynamic pick-up and delivery settings is very limited. To the best of the authors knowledge, Zhu et al. (2020) and Liu, He, and Max Shen (2018) are the only works that fit this description.

Zhu et al. (2020) predict arrival times by means of deep learning for a vehicle routing problem where requests come in dynamically. Uncertainty is present in bundling, 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 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. Their analysis indicates that their proposed deep learning architecture, inter alia, outperforms a GBDT approach w.r.t. prediction accuracy.

Liu et al. (2018) compare linear and piece-wise linear prediction models 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 bundling, 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. Analyzing the prediction models w.r.t to their accuracy, tractability and interpretability, they found that random forests yield the most accurate results but are computationally less tractable due to exponential runtimes and interpretable than linear models. Amongst the linear models, lasso regression obtained the smallest mean squared test error.

2.2 Arrival Time Estimation

With this section, we broaden the scope from arrival time estimations via supervised learning for dynamic pick-up and delivery problems to 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 approach. By *Route type*, the table identifies if arrival times are estimated for either single origin-destination pairs or sequences of them. By *Uncertainties*, the table refers to uncertain elements in the underlying problem settings. Uncertainty in request indicates that customer requests are not certainly known at the start of the problem and arrive dynamically. Uncertainty in travel and service times expresses itself through uncertain weather conditions, traffic congestion and individual challenges when serving customers (e.g. parking or waiting times). Uncertainty in processing occurs when

For single origin-destination problems, significant amount of work where travel times are estimated via supervised learning has been done for 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 and ANNs perform comparably good and outperform prediction methods which are based on historical average values or most recently obtained travel time information. 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) use a GBDT model based on manually selected travel time features and traffic state related variables. In their study, GBDTs outperform 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. Although 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)).

3 Model

Vehicle veV Node peP, deD actions a (consist of type, time t, len(action)) Routes 0 –> consists of actions

State (time, planned route, updated route)

4 Solution Approach

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 Motivation

We motivate this paper by answering the following questions: Why is machine learning useful and widely-common for esitmation tasks? Which technical problems are encountered when using other approaches? Which machine learning algorithms are appropriate for our problem? Why bother comparing them?

4.2 Algorithms

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

Supervised learning models receive a finite sequence $S = \{(x, y)|x \in X, y \in Y\}$ of length $N \in \mathbb{N}$ as inputs where $x_i \in X$ with $i \in N$ is an **observation** and $y_i \in Y$ its corresponding **target**. They aim for the approximation of an **hypothesis** $h : X \rightarrow Y$ that describes the underlying patterns of X accurately w.r.t. a **loss function** $L : Y \times Y \rightarrow \mathbb{R}$ and predicts a target of a given observation. Loss functions represent the deviance of $h(x_i)$ for a given x_i and its correspondent target y_i by which we can measure how accurate the predictions of a model are. An optimal approximation function h_{opt} is obtained by finding the minimal loss. h is then used to predict any $y_i \in Y$ for a corresponding $x_i \in X$.
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 Y, classification finds its use in the prediction of discrete Y. In machine learning terms, the arrival time estimation problems classifies as a regression task since we are estimating continuous time points. 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**.

4.2.1 GBDT

4.2.2 Random Forest

- Example - Formal Definition

4.2.3 Neural Network

- Example - Formal Definition

4.3 Feature Engineering

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

4.4 Evaluation

- Use MSE (and maybe more criteria) etc. to derive infos about accuracy - Hyperparameter sensitivity analysis (Random Search) for robustness - Variations in data set for robustness –> Noise einführen –> Anzahl der Trainingsdaten –> Andere Features - Time for performance

5 Computational Study

5.1 Experimental Design

5.2 Parametrization of Methods

- Different subsets of samples w.r.t size

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.
7

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

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.*
3

Huang, H., Pouls, M., Meyer, A., & Pauly, M. (2020). *Travel time prediction using tree-based ensembles.*
4, 5

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, 5

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
3

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
4

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
3

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
4

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
7

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
3