Linköpings universitet | Institutionen för datavetenskap
Examensarbete på avancerad nivå, 30hp | Datateknik
202018 | LIU-IDA/LITH-EX-A--2018/001--SE

Arrival Time Prediction On Buses in Linköping

–

Max Lund, Simeon Jackman, Per Lindström, Elias Alesand, Jakob Lundberg, Sebastian Callh

Handledare : Matthias Tiger
Examinator : Cyrille Berger

Linköpings universitet
SE–581 83 Linköping
013-28 10 00 , www.liu.se

Sammanfattning

Abstract.tex

Författarens tack

Acknowledgments.tex

Innehåll

1 Introduction

This report provides information regarding prediction of bus arrival times. It is part of *TDDE19 Advanced Project Course - AI and Machine Learning*.

1.1 Terminology

We will use the following terminology in this report:

- **Bus line:** Several data points from the same bus driving once
- **Trajectory:** Several data points from the same bus driving once
- **Journey:** All data from a bus driving and entire bus line from start to end
- **Segment:** All data points between two bus stops in a journey
- **Data point:** Term to be used to refer to the actual data

1.2 Problem

The problem at hand is to do bus time arrival prediction on the dataset provided by Östgötatraffiken.

1.3 Goals

- Literature study to find out about viable methods
- Pre-process data
- Predict bus arrival times using a simple baseline method
- Predict bus arrival times using Gaussian processes
- Predict bus arrival times using a neural network
- Compare solutions among each other

1.4 Approach

In this report bus time arrival prediction will be studied using a simple statistical baseline method, Gaussian processes, Neural networks and . The theory behind the solutions will be discussed in depth in chapter 2 and the methodology will be discussed in chapter 3.

The solutions will be compared by their mean absolute percentage error on identical predictons (MAPE).

1.5 Delimitations

Since generalizing our models completely is beyond the scope of this project, we had to simplify certain procedures. We have only examined few bus lines. Spatial outliers and incomplete journeys within those routes have been removed to reduce the complexity of the models needed to make predictions.

2 Theory

This chapter aims to explain concepts and methods that are used is our implementations.

2.1 Gaussion Processes

A Gaussian Process (*GP*) is, in statistical terms, a stochastic process such that any set of the variables have a multivariate normal distribution. Such a distribution is defined by a mean vector μ and covariance matrix σ, where every random variable spans one dimension. Viewed through the lens of machine learning, the GP serves as a non-parametric regression model.

Gaussian Process Regression

GP regression (also known as kriging) is a non-parametric generative model that builds on the assumption that all data points are drawn from a multivariate normal distribution. A GP f is consequently defined as $f \sim \mathcal{N}(0, \Sigma)$. Zero mean can be assumed without loss of generality and is done for mathematical convenience, but in practice μ could be any suitable mean vector. The covariance matrix Σ is thus the only free parameter to be chosen. Unfortunately, a covariance matrix over a continuous function would exist only in an infinite dimensional space, which is indeed problematic. However, the spaces we are interested in have an inner product corresponding to some covariance function $c(x, x') = cov(f(x), f(x'))$ for the process f and data points x, x', and this function can be kernelised as $k(\theta, x, x') = cov(f(x), f(x'))$ for some valid kernel function k and its hyperparameters θ. It is consequently possible to compute Σ given a set of data points and a kernel function, circumventing the need to construct infinite dimensional spaces.

Thanks to the kernel trick the choice of parameters is consequently not a covariance matrix Σ but a kernel $k(\theta, x, x')$ and its hyperparameters θ. This choice represent our prior over f, and in particular it represents how smooth we believe f is by imposing certain covariance on $f(x)$ for nearby points. This is the key idea of a GP regression and what makes it very flexible: It allows us to specify a prior over f as a function of x, without even knowing f.

2.2 Artificial Neural Networks

Artificial neural networks have shown to to be useful when predicting travel times due to their ability to model nonlinear relationships between features [5][1]. This section introduces the parameters that have been considered when creating artificial neural networks for this report.

Activation Functions

The purpose of the activation function is to compute the hidden layer values [7]. Two of the most popular activation functions are the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

and the rectifier function:

$$f(x) = max\{0, x\} \tag{2.2}$$

Using the rectifier function leads to less computationally complex learning than when using the sigmoid function although problems can occur where the backpropagation is blocked by a "dead" neuron due to the hard saturation at 0 [6].

Loss Functions

For an artificial neural network to be able to update its weights it needs a loss function that should be minimized in the case of gradient descent [7]. Some examples of common loss functions are **mean squared error (MSE)**, **mean absolute error (MAE)** and **mean absolute percentage error (MAPE)**. A problem with MAPE is that when the true value is 0 the function is undefined [3]. MSE gives a larger weight to outliers since the metric has an exponential relation to the error.

Hidden Layers and Neuron Count

The more hidden layers and neurons a network has the more computationally complex the learning becomes. Therefore you should not use a model that is more complex than the problem in question requires. Problems that involve large amount of input features like the *ImageNet* contest has been shown to be a case where deep neural networks perform well [8]. However, it has been shown that virtually any function can be approximated with two hidden layers given there are enough neurons [4].

2.3 Kalman Filters

Kalman filtering is an iterative algorithm used to increase measurment accuracy by considering multiple consecutive data points observed over time. It can be used in real time as new measurements are made. Kalman filtering has been shown to be useful when predicting travel times [9]. It has also been used together with artificial neural networks with promising results [2].

3 Method

3.1 Pre-processing

We have used jupyter notebooks as the main way to develop our code to pre-process. Data operations such as filtering are done with the python pandas package. The data for this project was delivered by Östergötatrafiken. The whole dataset is over 300GB in size. To be able to feed the data into our algorithms preprocessing had to be done.

Data structure

The provided data consists of one file per day over 90 day. Each day has an approximate size of 5GB. Within the data there are over 20 event-types that represent the state of a bus during a day. For our project we have used four types of events and discarded data with other events. The four events are:

ObservedPositionEvent: Gets triggered every second, contains the GPS data of a given bus.

EnteredEvent: Gets triggered when the bus is within a certain distance to a bus-station. Is used to split the journey into segments.

JourneyStartedEvent: Gets triggered when the bus is assigned a new journey. Is used to determine which line a bus is currently serving.

JourneyCompletedEvent: Gets triggered when the bus has completed a journey. Is used as a flag to determine when a journey has ended.

For our work we have chosen two bus lines, a subset of the bus line three and bus line number eleven. Line three has been used as a basis, since we all know that bus and we could assure that there were no irregularities on that line. Line eleven had been chosen because it captures a lot of problems that need to be taken into consideration within our predictions. Such problems are GPS variance due to high buildings, high red light density, and sharp corners.

Detecting Segments

The bus journey is split in to several segments. We have used the *EnteredEvent* to detect when a bus is approaching a station and therefore also to split the segments. The first segment starts, when a *JourneyStartedEvent* triggers. Incomplete or faulty journeys have been discarded, since it would be very difficult to model.

3.2 Artificial neural networks

The neural network models were made using *Keras* on top of *Tensorflow*.

Baseline ANN model

A simple model was created to use as a baseline model that can be used to compare other neural network models to. This model predicts the time it will take to travel to the next bus stop. As input this model use time of day normalized to a value in the range [0,1] and the segment for which the observation has been made. The segment input is one-hot encoded meaning that there is an input for each segment in the journey which all have a value of 0 except for the segment of the observation which has the value 1. The network has one fully connected hidden layer with 13 nodes and an output layer with one node. The network uses the *relu* activation function. This model predicts the time in seconds it will take to travel the whole segment. To get a prediction of the time to the next bus stop you need to subtract the actual known time travelled since the previous bus stop from the output of the neural network.

4 Results

This chapter presents the results. Note that the results are presented factually, striving for objectivity as far as possible. The results shall not be analyzed, discussed or evaluated. This is left for the discussion chapter.

In case the method chapter has been divided into subheadings such as pre-study, implementation and evaluation, the result chapter should have the same sub-headings. This gives a clear structure and makes the chapter easier to write.

In case results are presented from a process (e.g. an implementation process), the main decisions made during the process must be clearly presented and justified. Normally, alternative attempts, etc, have already been described in the theory chapter, making it possible to refer to it as part of the justification.

5 Discussion

This chapter contains the following sub-headings.

5.1 Results

Are there anything in the results that stand out and need be analyzed and commented on? How do the results relate to the material covered in the theory chapter? What does the theory imply about the meaning of the results? For example, what does it mean that a certain system got a certain numeric value in a usability evaluation; how good or bad is it? Is there something in the results that is unexpected based on the literature review, or is everything as one would theoretically expect?

5.2 Method

This is where the applied method is discussed and criticized. Taking a self-critical stance to the method used is an important part of the scientific approach.

A study is rarely perfect. There are almost always things one could have done differently if the study could be repeated or with extra resources. Go through the most important limitations with your method and discuss potential consequences for the results. Connect back to the method theory presented in the theory chapter. Refer explicitly to relevant sources.

The discussion shall also demonstrate an awareness of methodological concepts such as replicability, reliability, and validity. The concept of replicability has already been discussed in the Method chapter (3). Reliability is a term for whether one can expect to get the same results if a study is repeated with the same method. A study with a high degree of reliability has a large probability of leading to similar results if repeated. The concept of validity is, somewhat simplified, concerned with whether a performed measurement actually measures what one thinks is being measured. A study with a high degree of validity thus has a high level of credibility. A discussion of these concepts must be transferred to the actual context of the study.

The method discussion shall also contain a paragraph of source criticism. This is where the authors' point of view on the use and selection of sources is described.

In certain contexts it may be the case that the most relevant information for the study is not to be found in scientific literature but rather with individual software developers and open source projects. It must then be clearly stated that efforts have been made to gain access to this information, e.g. by direct communication with developers and/or through discussion forums, etc. Efforts must also be made to indicate the lack of relevant research literature. The precise manner of such investigations must be clearly specified in a method section. The paragraph on source criticism must critically discuss these approaches.

Usually however, there are always relevant related research. If not about the actual research questions, there is certainly important information about the domain under study.

5.3 The work in a wider context

There must be a section discussing ethical and societal aspects related to the work. This is important for the authors to demonstrate a professional maturity and also for achieving the education goals. If the work, for some reason, completely lacks a connection to ethical or societal aspects this must be explicitly stated and justified in the section Delimitations in the introduction chapter.

In the discussion chapter, one must explicitly refer to sources relevant to the discussion.

6 Conclusion

This chapter contains a summarization of the purpose and the research questions. To what extent has the aim been achieved, and what are the answers to the research questions?

The consequences for the target audience (and possibly for researchers and practitioners) must also be described. There should be a section on future work where ideas for continued work are described. If the conclusion chapter contains such a section, the ideas described therein must be concrete and well thought through.

Litteratur

[1] Mansur As och Tsunenori Mine. ”Dynamic Bus Travel Time Prediction Using an ANN-based Model”. I: *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*. IMCOM ’18. New York, NY, USA: ACM, 2018.

[2] Cong Bai, Zhong-Ren Peng, Qing-Chang Lu och Jian Sun. ”Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes”. I: *Computational Intelligence and Neuroscience* 2015, Article ID 432389 (2015).

[3] Zhuo Chen och Yuhong Yang. ”Assessing Forecast Accuracy Measures”. I: (2004). URL: https://www.researchgate.net/publication/228774888_Assessing_forecast_accuracy_measures.

[4] Howard B. Demuth, Mark H. Beale, Orlando De Jess och Martin T. Hagan. *Neural Network Design*. 2nd. USA: Martin Hagan, 2014. ISBN: 0971732116, 9780971732117.

[5] Wei Fan och Zegeye Gurmu. ”Dynamic Travel Time Prediction Models for Buses Using Only GPS Data”. I: *International Journal of Transportation Science and Technology* 4.4 (2015), s. 353–366.

[6] Xavier Glorot, Antoine Bordes och Yoshua Bengio. ”Deep Sparse Rectifier Neural Networks”. I: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Utg. av Geoffrey Gordon, David Dunson och Miroslav Dudík. Vol. 15. Proceedings of Machine Learning Research. Fort Lauderdale, FL, USA: PMLR, nov. 2011, s. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[7] Ian Goodfellow, Yoshua Bengio och Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[8] Alex Krizhevsky, Ilya Sutskever och Geoffrey E. Hinton. ”ImageNet Classification with Deep Convolutional Neural Networks”. I: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, s. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[9] Jiann-Shiou Yang. ”Travel time prediction using the GPS test vehicle and Kalman filtering techniques”. I: *Proceedings of the 2005, American Control Conference*. ACC ’05. Portland, OR, USA: IEEE, 2005.