



# Real-time travel time prediction models in routing for car navigation applications

**P. GAWEŁ<sup>a,b</sup>, K. DEMBCZYŃSKI<sup>a</sup>, W. KOTŁOWSKI<sup>a</sup>, A. JASZKIEWICZ<sup>a</sup>**

<sup>a</sup> POZNAŃ UNIVERSITY OF TECHNOLOGY, Piotrowo 2, 60-965 Poznań, Poland

<sup>b</sup> NAVIEXPERT SP. Z O.O., Dobrzyckiego 4, 61-692 Poznań, Poland

EMAIL: przemyslaw.gawel@cs.put.poznan.pl

## ABSTRACT

We consider the problem of using real-time floating car data to construct vehicle travel time prediction models meant to be used as input to routing algorithms for finding the fastest (time-shortest) path in the traffic network. More specifically we target the on-line car navigation systems. The travel time estimates for such a system need to be computed efficiently and provided for all short segments (links) of the roads network. We compare several fast real-time methods such as last observation, moving average and exponential smoothing, each combined with a historical traffic pattern model. Through a series of large-scale experiments on real-world data we show that the described approach yields promising results and conclude that specific prediction function form may be less important than a proper control of bias-variance trade-off (achieved by historical and real-time models combination). In addition, we consider two different settings for testing the prediction quality of the models. The first setting concerns measuring the prediction error on short road segments, while the second on longer paths through the traffic network. We show the quality and model parameters vary depending on the assessment method.

**KEYWORDS:** fcd, travel time, prediction, navigation, routing

## 1. Introduction

Travel time estimation and prediction is a fundamental and important part of many traffic-related systems, but it can take many different forms depending on the available data or desired application. In this paper, we focus on constructing travel time prediction models for routing, or more specifically for online personal car navigation systems that aim to find the fastest (time-shortest) path between any two points in the traffic network. The input to proposed models will also naturally consist of floating car data (FCD), i.e. the information originating from moving vehicles equipped with GPS devices (e.g. cars with personal navigation).

The structure of the above problem differs significantly from the traditional approaches to travel time estimation, where data is

commonly provided by stationary sensors e.g. loop detectors [7] or similar [4]. Contrary to stationary sensors, the FCD does not need expensive infrastructure and can cover the whole roads network (as opposed to selected parts), but the observations are randomly and unevenly distributed in space and time, as a result this kind of data requires a largely different construction of models, even if the applied mathematical methods seem similar.

Moreover, the majority of the current research does not focus on the routing-compatible single road segments (sometimes called links) and the entire traffic network, instead researching e.g. single paths [7] or freeways [1, 6]. There are generally few papers targeting routing applications and FCD explicitly and even those usually don't focus on the broad (both long-term and short-term) scope of prediction – e.g. in a recent paper targeting travel times for city logistics [3] only the long-term traffic patterns are considered.

A variety of methods can be applied to the travel time prediction for routing. In our approach we analyse several methods of different complexity. The simplest model is based on the *last observation*. The two other more complex models are the *moving average* and *exponential smoothing*. These models can optionally be mixed with a historical (static) traffic patterns model (similar to [2] or [3]) to reduce the variance of the most recent observations and augment the long-term prediction. This approach can be similar to some other proposed solutions – e.g. a complex system [5] incorporating historical and real-time model, as well as expert rules and external event inputs. Another similar approach concerns supplementing time series models with historical data [8]. In relation to the above solutions we try to achieve similar goals with noncomplex models, naturally formulated for the continuous time domain (as opposed to discrete time series).

Our empirical study thoroughly tests all the methods on a large set real-world data (a traffic network of an entire city) and finds that the applied approach yields promising results. We additionally discuss the relation between prediction quality and the model complexity as well as the optional mixing with a historical traffic patterns model.

We also consider two different techniques of assessing the prediction quality. One technique concerns measuring the prediction error on short road segments (on which we predict), while the other uses longer paths through the traffic network (which is closer to the route planning application). We discuss the differences between them.

## 2. Problem statement

The task is to find a function  $\hat{y}_{st} = f(s, t) \in R^+$ , that accurately predicts an actual travel time  $y_{st} \in R^+$  for each individual short road segment  $s \in S$  in a given time point  $t \in R^+$ , where  $R^+$  is the set of nonnegative reals. By this we want to model the travel times in the roads network on a high level of granularity. The accuracy of a single prediction  $\hat{y}$ , having a true value  $y$ , is measured by the squared error loss  $L(y, \hat{y}) = (y - \hat{y})^2$ , and for the purpose of model evaluation, this loss is preferably measured on longer vehicle paths than on short road network segments.

## 3. Models

The predictions come from a composition of two models. The first one, a static model, predicts daily, weekly, and overall periodical trends in the traffic (e.g. “at every Tuesday morning, on a road segment in the city centre, the traffic is low”) using a large historical data set (several months of past data). This constitutes its strength (stability and long-term prediction), but also its weakness (no adaptation to current traffic conditions). Therefore, the static model is combined with a dynamic model, which exploits recent real-time observations. The dynamic model targets the short-term prediction by fitting to the current traffic situation (deviation from the static model), but it is unable to predict in the long term on its own, and can have a much higher variance.

### 3.1 Static model

For the purpose of this research we will use a simple base static model that will be denoted as  $f_b(s, t)$ . Let us consider a model that divides the day into four general time periods: the morning rush, the midday, the afternoon rush and the night. Then, it predicts using mean travel time  $\bar{y}_{sp}$  computed for each road segment  $s$  and time period  $p$  over a set of historical data. This model is a simple ‘step function’ in the time domain for a particular segment. It is easy to compute, has a compact representation (only four travel times) and the prediction consists only of determining the correct time period. To obtain the final static model, having a more natural smooth transition between time periods, we additionally apply a tricube (fast, Gaussian-like) smoothing function.

The above model is one of many possible static models that could server as a stable base for the following dynamic models and it could be easily replaced by slightly different models serving the same purpose (e.g. [3] or [2]). There are even commercially sold equivalents, such as TomTom Speed Profiles or NAVTEQ Traffic Patterns. We will not discuss this model in much detail as part of this particular research.

### 3.2 Dynamic models

A dynamic model  $f_d(s, t, Y_{sr})$  depends only on the latest real-time travel time observations  $Y_{sr} = \{y_{st_1}, y_{st_2}, \dots, y_{st_n}\}$  where  $n$  is number of latest observations for segment  $s$ . The set of recent observations in the time window of the length  $w$  from  $t_0$  can be defined as:  $Y_{swt_0} = \{y_{st_i} \in Y_{sr} : t_0 - t_i < w\}$ .

Each dynamic model will also have two parameters (discussed in the next section) that can be tuned: either  $w$  or  $T$ , and a parameter  $\lambda$  that can be optionally used to enable mixing with a base static model  $f_b(s, t)$ .

The last observation model uses the latest observed travel time on a particular segment as the predicted value, which can be effective. e.g.. for a high congestion, where subsequent vehicles move with a similar low velocity. The model can be written as:

$$f_{lo}(s, t_0, Y_{sr}) = \frac{y_{st_{\max}} + \lambda f_b(s, t_0)}{1 + \lambda}, \quad t_{\max} = \max_{y_{st_i} \in Y_{swt_0}} t_i \quad (1)$$

Additionally if  $Y_{swt_0} = \emptyset$  the prediction falls back fully to the base static model  $f_b(s, t)$ .

Since the last observation model predicts using a single observation and no averaging is involved, the variance of the prediction can be high. Therefore, the moving average model uses the mean of several latest observed travel times on a particular segment, thus reducing variance and potentially working better for more noisy cases – such as a traffic lights or lower data density.

The model can be written as:

$$f_{ma}(s, t_0, Y_{sr}) = \frac{\left( \sum_{y_{st_i} \in Y_{swt_0}} y_{st_i} \right) + \lambda f_b(s, t_0)}{|Y_{swt_0}| + \lambda} \quad (2)$$

The *exponential smoothing* model is usually defined recursively as a time series method, assuming that observations are available at regular time intervals. This cannot be applied directly when

dealing with floating car data incoming at random (often large) intervals or missing entirely. We thus use a version of exponential smoothing suited for the continuous time domain. Contrary to the moving average model, which uses a hard-threshold time window, the exponential smoothing method uses a weighted average of previous observations, with weights decaying at exponential rate. There is also a smooth exponential fallback effect. The model can be formally defined as:

$$f_{es}(s, t_0, Y_{sr}) = \frac{\left( \sum_{y_{st_i} \in Y_{sr}} g_i y_{st_i} \right) + \lambda f_b(s, t_0)}{\left( \sum_{y_{st_i} \in Y_{sr}} g_i \right) + \lambda} \quad (3)$$

where  $g_i = \exp(-(t_0 - t_i) / T)$  and the time constant  $T$  determines the speed of decay. Using this exponential decay function is very advantageous in terms of memory consumption and computational time, as we can (similarly to the recursive version of the formula) easily reuse the previous predictions, only performing a few multiplications and additions when a new observation appears or a prediction is needed.

### 3.3 Parameters

The interpretation of the parameters is quite straightforward. The time window is a fairly basic concept and the time constant has a similar meaning. Whereas in the case of a time window, every observation in the window could be thought of as having the weight of 1, and every observation outside of the window as having the weight of 0, in the case of exponential decay, the weight of observations decreases from 1 (at current time) to 0.37 after  $1T$ , 0.14 after  $2T$ , 0.05 after  $3T$  and below 0.02 after  $4T$ , reaching values close to 0 thereafter.

The parameter  $\lambda$  allows for mixing ('shrinking') with a stable default prediction (static model) and can be thought of as the number (or a sum of weights) of artificial observations that are added in the formula of the models. This parameter serves to control the bias-variance trade-off of the short-term prediction.

We optimize the parameters globally, by performing a simple grid search, checking each pair of parameters from the domains:  $T, w \in \{0.125, 0.250, \dots, 3.0\}$ ,  $\lambda \in \{2^{-8}, 2^{-7}, \dots, 2^4\}$ , where the unit of  $T$  and  $w$  is hours. These ranges were determined using the parameters interpretation, as well as by preliminary experiments. We note, that all parameter tuning is done on training data in order to get unbiased results on the testing data.

## 4. Experiments

We use floating car data covering the city of Poznań, one of the major cities of Poland with about half a million population, with surroundings (covering the area of more than 400km<sup>2</sup>). The observations span is more than a month – from 12th May 2012 to 16th June 2012. The data sample was provided by *NaviExpert* – an on-line personal car navigation company.

The input takes the form of travel time observations on individual short road segments at particular time instants. These observations

originate from vehicles equipped with GPS, either through navigation or monitoring devices. The data has been preprocessed and cleaned by *NaviExpert's* data processing algorithms (this includes map-matching and some outliers filtering). The roads map used in the experiments was the OpenStreetMap1 from May 2012.

In the experiments, the predictions were always evaluated on observations that had no chance to influence the models beforehand. We also assume a sensible latency  $T_L = 5$  min, to avoid using observations that would not yet enter the system in reality, and a retention time  $T_R = 6$  hours for the real-time data:  $\forall y_{st} \in Y_r : t_0 - T_R < t < t_0 - T_L$ , where  $t_0$  is the current time.

The data was also divided into subsets, to ensure reliable results. The test set ranging from 10th May 2012 to 16th June 2012 was used for the final models evaluation, while the rest of the data (from 9th May 2012 to 12th May 2012) was used as a learning set – to compute the historical traffic patterns and tune the parameters of the models.

Additionally, all of the data collected at late night hours – ranging from 11 p.m. to 4 a.m. are filtered out of the experiments, as potentially containing outliers. We also consider only the main roads of the city (skipping less important roads on the residential level) to better emphasize the cases where traffic congestion is possible and natural.

### 4.1 Results

The main results comparing the evaluated static and dynamic models can be seen in Table 1. The models are evaluated on observed vehicle paths (or parts thereof) – ranging from 0.5km to 5km, having an average length of about 2.8km. We constrain the maximum length of the vehicle paths to 5km by dividing them as needed, as we do not want to suffer from the variance resulting from severely different route lengths between vehicles. We use the root mean square error (RMSE), as well as the mean absolute error (MAE), which is less susceptible to outliers. The values are also accompanied (after the  $\pm$  sign) by statistical standard error for the obtained error measure values.

We also use a simplest segment mean model as the reference for percentage error; this model predicts using a constant historical mean travel value for each segment.

**Table 1. Total errors for models**

model	T w	$\lambda$	MAE[min]	MAE[%]	RMSE[min]	RMSE[%]
segment mean			1.5762 ± .0074	100.00	2.6632 ± .0361	100.00
time periods			1.4545 ± .0071	92.28	2.5074 ± .0348	94.15
last observation	1.000	0	1.3971 ± .0063	88.63	2.3017 ± .0188	86.43
last observation	2.625	1.000	1.2653 ± .0058	80.28	2.0965 ± .0221	78.72
moving average	0.875	0	1.3367 ± .0061	84.81	2.2222 ± .0207	83.44
moving average	0.875	1.000	1.2539 ± .0058	79.55	2.0972 ± .0232	78.75
exp. smoothing	0.375	0	1.2788 ± .0061	81.13	2.1771 ± .0220	81.75
exp. smoothing	0.250	0.125	1.2378 ± .0057	78.53	2.0696 ± .0231	77.71

mean path length = 2.837[km] number of vehicle paths = 83552

<sup>1</sup> Available at <http://www.openstreetmap.org/> under CC-BY-SA license, created by OpenStreetMap contributors

The time periods model yields a considerable improvement over the simple segment mean, but the accuracy of short-term prediction is naturally limited by the lack of real-time input data.

The dynamic models with mixing parameter  $\lambda = 0$  are purely real-time versions, not mixed with the static time periods model (falling back to the static segment mean only in the case of no real-time observations). This shows that the mixing with time periods model (by setting  $\lambda > 0$ ) really improves the quality, and the improvement is most prominent for the simple last observation model. We also see that the exponential smoothing is the best amongst this group of models, both with and without the time periods model. Nevertheless, the differences between the three methods are not that large, when we allow for  $\lambda > 0$  — especially in case of RMSE and its standard error the differences seem statistically negligible. This suggests that the ability to control the bias-variance trade-off of the model by mixing with a stable default (static model) may be more important than the choice of a specific formula for dynamic short-term forecasting.

Next we consider two different techniques of assessing the prediction quality, both of which seem sensible for our problem.

The first assessment technique concerns measuring the prediction error on short road segments (or links). These segments are the main unit of the roads network. The direct routing-compatible prediction also takes place on each individual segment.

The second technique (the one also applied in Table 1) uses longer paths through the traffic network. These paths are closer to the route planning application, as the potential users of the prediction model will be interested in itineraries much longer than single road segments. In this case the predicted travel time value is the sum of predictions made on individual segments that a given path consists of.

Table 2 shows two versions of exponential smoothing model. The first version ( $T = 0.25, \lambda = 0.125$ ) has its parameters optimized for the longer paths case while the second version ( $T = 0.5, \lambda = 1$ ) has parameters optimized with regards to the squared error on the short segments. We clearly see, that the optimal parameters can be very different — the case of longer paths calls for a more reactive and unbiased model, while the case of short segments requires a more conservative and less reactive approach.

**Table 2. Short segments and longer paths comparison**

		short segments		longer paths		
		d [km]	count	d [km]	count	
		0.131	1836443	2.837	83552	
model	T	$\lambda$	RMSE[min]	RMSE[%]	RMSE[min]	RMSE[%]
exp. smoothing	0.250	0.125	0.3458 ± .0024	91.79	2.0696 ± .0231	77.71
exp. smoothing	0.500	1.000	0.3362 ± .0026	89.26	2.1208 ± .0259	79.63

This shows that if we want a travel time prediction model to produce good prediction on longer paths, then we need to specifically optimize the error for these cases, even if we need to

predict on the more granular level of a single segment. For our main results Table 1 we have chosen to optimize and evaluate the models for the longer paths case, as it seems closer to the ultimate goal of computing fastest paths in routing algorithms, but further research into the subject of a proper model quality assessment technique may be beneficial.

## 5. Conclusion

In this paper, we considered a problem of travel time prediction for an on-line personal car navigation system and other routing applications. We introduced solutions suited to FCD, that combine static and dynamic models. We also performed exhaustive large-scale experiments on real-world data.

The experimental results have shown that simple and fast dynamic travel time prediction methods, applied on the level of single short road segments, can yield good results, especially when combined with a stable default such as the static traffic patterns model. It was also shown that it is important to choose the case for which we predict (e.g. longer paths or short road segments) and optimize the bias-variance trade-off accordingly.

The work can be extended e.g. by individualizing the parameters of models for single segments and by complementing the models with more specialized methods for detecting unusual traffic incidents, falling outside of the scope of natural randomness and congestion. Further research into the proper model quality assessment technique may also be interesting.

## Acknowledgments

This work has been supported by the Polish National Science Centre, grant no. N N519 441939.

## Bibliography

- [1] BILLINGS, D., YANG, J.: Application of the ARIMA Models to Urban Roadway Travel Time Prediction-A Case Study. In: IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC'06. vol. 3 (2006)
- [2] DEMBCZYŃSKI, K., KOTŁOWSKI, W., GAWEŁ, P., SZARECKI, A., JASZKIEWICZ, A.: Matrix factorization for travel time estimation in large traffic networks. In: Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 7895, pp. 500–510. Springer Berlin Heidelberg (2013)
- [3] EHMKE, J.F., MEISEL, S., MATTFELD, D.C.: Floating car based travel times for city logistics. *Transportation research part C: emerging technologies* 21(1), 338–352 (2012)
- [4] HIRAMATSU, A., NOSE, K., TENMOKU, K., MORITA, T.: Prediction of travel time in urban district based on state equation. *Electronics and Communications in Japan* 92(7), 1–11 (2009)
- [5] LEE, W., TSENG, S., TSAI, S.: A knowledge based real-time travel time prediction system for urban network. *Expert Systems With Applications* 36(3P1), 4239–4247 (2009)

- [6] VAN LINT, J.: Reliable real-time framework for short-term freeway travel time prediction. *Journal of Transportation Engineering* 132, 921 (2006)
- [7] VAN LINT, J., HOOGENDOORN, S., VAN ZUYLEN, H.: Accurate freeway travel time prediction with state-space neural networks under missing data. *Transportation Research Part C* 13(5-6), 347–369 (2005)
- [8] ZHU, T., KONG, X., LV, W., ZHANG, Y., DU, B.: Travel time prediction for float car system based on time series. In: *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*. vol. 2, pp. 1503–1508 (2010)