Utilizing data mining techniques to predict expected freeway travel time from experienced travel time

Hasan M. Moonam, Xiao Qin, Jun Zhang

PII: S0378-4754(18)30026-0
DOI: https://doi.org/10.1016/j.matcom.2018.01.006
Reference: MATCOM 4538

To appear in: Mathematics and Computers in Simulation

Received date: 9 October 2017
Revised date: 27 January 2018
Accepted date: 29 January 2018


This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Utilizing Data Mining Techniques to Predict Expected Freeway Travel Time from Experienced Travel Time

Hasan M. Moonam a,1, Xiao Qin a,*, Jun Zhang b,2
a Civil and Environmental Engineering, University of Wisconsin-Milwaukee, 3200 N Cramer St., Milwaukee, WI 53211-3314, United States
b Electrical Engineering & Computer Science, University of Wisconsin-Milwaukee

ABSTRACT

As the most important real-time traveler information, travel time can be either experienced or expected (i.e. to be experienced). When a vehicle completes a trip, the travel time refers to the experienced travel time. In contrast, when a vehicle starts its journey, the travel time is unknown but can be predicted, which is the expected travel time. Although the experienced travel time is termed as the real-time travel time, a traveler may encounter a somewhat different travel time (from expected travel time) due to the changing traffic conditions. Therefore, expected travel time needs to be predicted. In this study, the expected travel time was predicted from the experienced travel time using the data mining techniques such as k-nearest neighbor (k-NN), least squares regression boosting (LSBoost) and Kalman filter (KF) methods. After comparing the performances of KF to corresponding modeling techniques from both link and corridor perspectives, it is concluded that the KF method offers superior prediction accuracy in a link-based model. Moreover, the effect of different noise assumptions was examined and it is found that the steady noise computed from the full-dataset had the most accurate prediction. A data processing algorithm, which processed more than a hundred million records reliably and efficiently was also introduced.

Keywords: experienced and expected travel time, arrival and departure time based travel time, travel time prediction, data mining, Kalman filter, modeling Kalman filter noise, K-nearest neighbor method, Boosting, LSBoost.

1. Introduction

Travel time is an important component of Advanced Traveler Information Systems (ATIS), as it is a key factor for travelers who are faced with non-recurring congestion (Khattak et al., 1996). Aside from measuring transportation system performance, travel time has been used to predict future travel time and traffic state, which help the traffic operations room in versatile ways. Amongst all available techniques, Bluetooth has emerged as one of the fastest growing data collection technologies whose market share is continuing to rise, mainly due to its cost effectiveness (Blogg et al., 2010, Moghaddam and Hellinga, 2013). Bluetooth is a probe-based...
(Puckett and Vickich, 2010) Automatic Vehicle Identification (AVI) technique used for collecting travel time data. Each Bluetooth device contains a unique electronic identifier known as a Media Access Control (MAC) address/identifier. Devices with MAC addresses that are in range can be logged as long as a simple antenna is mounted adjacent to the roadway. Travel time and corresponding traffic speed can be estimated by the timestamp difference between two consecutive stations with matching MAC addresses (Bachmann et al., 2013).

Travel time is measured by the time elapsed when a traveler moves between two distinct spatial positions (Carrion and Levinson, 2012). Over a decade, various studies have attempted to define travel time estimation (e.g. instantaneous travel time, experienced travel time (Xiao et al., 2014), and predicted travel time (Bhaskar et al., 2011)), but the definitions lack clarity, which has created confusion and inconsistency in data collection and analysis (Bhaskar et al., 2011, Toppen and Wunderlich, 2003). Recently, one study made clear the distinction between arrival time-based link travel time and departure time-based link travel time (Kim et al., 2009).

Arrival time-based link travel time (ATT) and departure time-based link travel time (DTT) have two different estimation algorithms. ATT refers to the travel time associated with arrival at the destination, while DTT refers to the travel time associated with departure from the origin. In practice, ATT is the experienced travel time \( t_{\text{expn}} \) that is calculated using arrival and departure times of the vehicles when both are available. On the other hand, DTT is the expected travel time \( t_{\text{expt}} \) that is predicted at the time of departure when arrival time is unavailable. To get ATT or \( t_{\text{expn}} \) and DTT or \( t_{\text{expt}} \), assume that two vehicles \( V_1 \) and \( V_2 \) start at 8:30am and 9:00am from point A, respectively. If the assumed clock time is now 9:00am and the first vehicle \( V_1 \) has just arrived at point B. Then, the arrival time based travel time for link AB, \( ATT_{AB@9:00AM} = 30\text{mins} \) (based on the experienced travel time of \( V_1 \)). The departure time based link travel time, \( DTT_{AB@9:00AM} = \text{unknown} \). If the arrival time for \( V_2 \) at point B could be predicted (say, 9:25am), the departure time based travel time would be, \( DTT_{AB@9:00AM} = 25\text{mins} \) (based on the expected travel time of \( V_2 \)). Since the DTT at 9:00am is unavailable until a later time, i.e., until the \( V_2 \) travels the link AB, it is understandable that the DTT at 9:00am requires a prediction of travel time. Fig. 1 illustrates the concept of estimating ATT or \( t_{\text{expn}} \) and DTT or \( t_{\text{expt}} \) for a route considering multiple vehicles:

Practitioners usually treat ATT (or \( t_{\text{expn}} \)) as the travel time due to the lack of available DTT (or \( t_{\text{expt}} \)); however, this reported ATT is one-step (step interval = travel time) earlier than the actual travel time to be experienced (DTT) by drivers. Although ATT and DTT differ slightly in a free-flow condition, the difference can sharply escalate at the onset and end of traffic congestion. ATT usually lags behind DTT during a transition of traffic state, and the difference
starts to decrease when the traffic state becomes stable. Information on travel time during a transitional state, as opposed to a stable state, is more important to travelers. Ideally, the travel time should be the predicted travel time that will be experienced by a traveler, or DTT. This predicted travel time also helps ensure proper and proactive operations and management of traffic in a network. Unfortunately, few studies have distinguished between DTT and ATT or attempted to estimate DTT (Kim et al., 2009).

The overarching goal of this study is to develop a comprehensive model for short-term freeway travel time prediction using Bluetooth data. A dynamic filtering algorithm was proposed to accurately estimate ATT and thus, reliably predict DTT. An efficient computer algorithm was developed to process, refine, and integrate a massive Bluetooth dataset, which filtered the travel time. Finally, prediction algorithms were examined to predict DTT from real-time ATT, and the better performance of the proposed prediction was observed.

2. Literature review

Travel time data are subject to outliers. The main purpose of outlier detection algorithms is to detect extreme travel times that result from sampling bias. Fixed-range outlier filtering methods are not suitable for travel time filtering due to local travel time turbulences, especially when they occur at the onset or end of congestion. To avoid imposing arbitrary fixed-bound, researchers have introduced moving average speed based lower and upper-bound (Haghani et al., 2010) and data-driven real-time adaptive-bound methods (Dion and Rakha, 2006). (Dion and Rakha, 2006) incorporated a few simple yet significant alterations in their proposed adaptive method, which offers an alternative to conventional algorithms like percentile, deviation, and traditional (modified) z- or t-statistical test (Liu, 2008, Clark et al., 2002). The main alteration includes expanding the data validity window when three consecutive observations fall either above or below (same side) the window. While this key adjustment helps capture sudden changes in travel time trends, it is prone to the inclusion of extreme outliers, and therefore compromises the accuracy of travel time estimation. In response to (Dion and Rakha, 2006) method, (Moghaddam and Hellinga, 2014) proposed a proactive method that uses a pattern recognition model, which showed superior performance. But the author acknowledged that the performance of outlier detection algorithms cannot be objectively quantified when the algorithms are applied to field data. Appropriate estimation of travel time is possible only when an effective outlier filter is used. Many studies have examined accurate estimation of travel time in a real-time fashion (Dion and Rakha, 2006, Skabardonis and Geroliminis, 2005, Moghaddam and Hellinga, 2014, Rice and Van Zwet, 2004). In most cases, the sophistication of filtering algorithms to maximize the accuracy led to a certain level of complexity in real-time applications. Therefore, a simplified version of these proposed algorithms is preferred.

Broadly, travel time prediction methods can be classified into the classical approach (Oda, 1990) including statistical (Rice and Van Zwet, 2004) and time series models (Al-Deek et al., 1998, Hamed et al., 1995), and the data mining approach (Vlahogianni et al., 2014, Zheng and Van Zylen, 2013, Zhang et al., 2014, Wu et al., 2004, Myung et al., 2011). Due to the instability of traffic states, most classical approaches have shown to be incapable of better prediction, especially with regard to structured and unstructured data (Vlahogianni et al., 2014). Therefore, advanced data mining methods have become popular to predict travel time. As such, (Jenelius
and Koutsopoulos, 2017) proposed a multivariate probabilistic principal component analysis method that predicts travel time based on the expected distribution of link travel times. The method provides superior results to the k-NN method. (Zhong et al., 2017) introduced an online travel time prediction system without sacrificing computational efficiency. The system adopts functional principal component analysis framework and utilizes historical and real-time travel time data to predict link travel time. (Zhan et al., 2015) applied a two-component generalized autoregressive conditional heteroskedasticity (GARCH) model that captures trend and seasonal components to improve prediction results. (Fei et al., 2011) employed a Bayesian inference-based dynamic linear model (DLM) to predict online short-term travel time. (Sumalee et al., 2013) estimated dynamic stochastic journey time distribution and predict travel time based on stochastic cell transmission model. (Zhan et al., 2013) extracted travel time data from origin-destination dataset by minimizing the least squared error between the observed and expected path travel times. (Zou et al., 2014) exploited a space-time diurnal method to predict travel time that considers spatial and temporal correlation and diurnal pattern of travel times. In addition to vehicle trajectory-based methods, (Celikoglu, 2013) used flow model to predict travel time that not only increases accuracy but also reduces computational complexity. Since data mining approaches fit easily with massive datasets, researchers have also applied neural networks (Zheng and Van Zuylen, 2013), fuzzy and evolutionary techniques (Zhang et al., 2014), support vector regression (Wu et al., 2004), and the k-nearest-neighbor (k-NN) (Myung et al., 2011) model to directly or indirectly predict travel time. According to Myung et al., the use of non-representative samples to train the artificial neural-network (ANN) model may lead to a non-negligible error in prediction (Myung et al., 2011). These data mining approaches require representative samples (Smith et al., 2002) and sometimes suffer from a lack of interpretability and transferability. On the other hand, simple methods (e.g. instantaneous, historic average and clustering over specific days) exhibit low accuracy (Van Hinsbergen et al., 2007). Although LSBoost, a data mining technique, has been widely used (Jiang, 2001a, Jiang, 2001b, Barutçuoğlu and Alpaydın, 2003, Jiao et al., 2006, Darwish, 2013) in different fields for many years, only recently been popular in transportation studies such as perception reaction time (Elhenawy et al., 2017), congestion duration (Ghosh et al., 2016), freight flow (Moscoso- López et al., 2016) and emission (Oduro et al., 2015) prediction.

KF, an optimal recursive data processing algorithm, has been widely used with various modifications (e.g. adaptive KF (Guo et al., 2014) and extended KF (Liu et al., 2006)) in several studies including those on travel time prediction (Chien and Kuchipudi, 2003, Nanthawichit et al., 2003, Yang, 2005, Chen and Chien, 2001). KF incorporates all information that can be provided and processes all available measurements to estimate the current value of the variables of interest (Maybeck, 1990). The KF method has two components: process/system/state-space and measurement/observation. Nanthawichit et al. formed their state-space equation by declaring traffic density and space mean speed as state variables, developed their observation equation by declaring traffic volumes and spot speeds as observation variables (Nanthawichit et al., 2003). Chen and Chien used travel time as the input variable in both of these equations; previous step travel time was multiplied by a transition matrix to obtain the state update equation (Chen and Chien, 2001). A similar study was conducted using field data (Chien and Kuchipudi, 2003) rather than simulated data (Chen and Chien, 2001). Despite promising results, these studies lack details about sources of process and measurement (variables') values.
Researchers (Chen and Chien, 2001, Chien and Kuchipudi, 2003, Nanthawichit et al., 2003, Yang, 2005, Chen and Rakha, 2014) have modeled the state-space as a linear system to which KF was applied. The KF model uses both a priori (derived by state-space/process) and the observation of the same timestamp to get a posterior by using the update equation. Within this procedure, the model combines all available observations and prior knowledge of a system in a way that statistically minimizes errors (Maybeck, 1990). The model can be directly applied to signal processing and control systems (when observations are available) in order to filter out the noise. KF has also been widely applied to time series forecasting with a state-space model (Durbin and Koopman, 2012, Hamilton, 1994, Harvey, 1990) where the update equation (Commandeur and Koopman, 2007, Durbin and Koopman, 2012) clearly states a time latency; the next step is predicted using the observations from the current step. Hence, predicting DTT from ATT is more advantageous over a single source time series data models.

3. Data preparation and reduction

Bluetooth data contains three variables: the MAC ID of the detector, MAC IDs of the detected devices, and the detection timestamp. In spite of a simple data format, a complex processing algorithm is required to produce the final dataset from the source, which stores the entire network data in a single table. For a logged MAC ID, recorded timestamps at two consecutive stations are processed to estimate travel time and the corresponding traffic speed. The detailed description of the processing algorithm is beyond the scope of this paper; therefore, a brief description is included below with a limited description of the data characteristics.

The selected study area consists of a 62.8-mile long route, or approximately 47.5 miles on I-90 and the remaining on the Beltline Highway in Madison, Wisconsin. The route is equipped with 41 unequally spaced Bluetooth stations, resulting in 40 links. The first 21 links are on I-90, the 22nd link is on both corridors, and the remaining links are on the Beltline. The spacing varies from 1.3-3.4 miles on I-90 and 0.4-1.3 miles on the Beltline Highway. Forty-seven days’ worth of data (11/16/2015-01/01/2016) containing more than 100 million records was collected from traffic in both directions. Half of the records were from outside the study-area. Each station of the one hundred stations selected captured around one million records for 47 days, or 67,680 minutes. However, a large portion (approx. three-fourths) of the data are either corrupted or contaminated due to multiple detections and unsuccessful detections (i.e. not detected in two consecutive stations). Fig. 2 shows the complete procedure of data processing.
A Bluetooth station usually detects a Bluetooth device in its range more than once. The number of such detections can increase significantly due to planned or unplanned slowing down/stopping of vehicles. A general inspection of the dataset revealed that such detections usually vary two to four times. Oracle queries helped clean up the multi-detection, resulting in the total number of records decreasing from 105 million to 26 million. The data was then separated by each station for the selected routes, further reducing the records to 10 million. Unsuccessful detections were automatically ignored due to the vehicle’s detection timestamps from two adjacent stations. Travel times were calculated. Next, the reduced dataset of 8 million samples was processed through a robust Java-based pre-processing module that investigated each record individually and cleaned all redundant records based on the following principle:

*Two detections of a vehicle at a station (STA$_1$) are valid separate detections if the vehicle is detected at least once at its upstream station (STA$_2$) within the time gap of two detections at STA$_1$.*

For example, a vehicle detected on 08:59am, 09:01am and 09:08am at STA$_1$, and on 09:04am at STA$_2$. Detection on 09:01am is redundant since there is no detection at upstream station STA$_2$ in between 08:59am and 09:01am. In addition, neither 09:01am nor 09:08am is a redundant detection since there is a detection at upstream station STA$_2$ on 09:04am. Note that the Bluetooth stations were capturing both directions of traffic and the vehicle used as an example made a U-turn/return-trip. The pre-processed dataset of 7 million records was further processed to filter outliers. Finally, a Java-based programming module produced the travel time and speed data using the outlier-filtered data. Since travel direction is pertinent to travel time, this study used northbound data.
4. Methodology

The methodology section details the algorithms for outlier filtering, ATT, DTT, travel speed estimation, and travel time prediction.

4.1. Outlier Filtering

A preset upper and lower boundary helped filter out the outliers. The following equation defines the lower boundary:

\[ t t_{lowr} = t t_{ff} / 2 \]  

(1)

where, \( t t_{lowr} \) and \( t t_{ff} \) stand for lower bound and free flow travel time, respectively. A vehicle was considered to be an outlier if its speed exceeded more than double the posted speed limit.

A dynamic validation window works best in outlier filtering (Dion and Rakha, 2006). The upper boundary is defined by the following equation:

\[ t t_{uppr} = t t_{e} + n \cdot \sigma_{e} \]  

(2)

where \( t t_{uppr} \), \( t t_{e} \) and \( \sigma_{e} \) are upper bound, expected travel time, and expected standard deviation of travel time (samples), respectively. \( n = 1, 2, 3, .... \)

4.2. Speed Estimation

The following equation estimates the space mean speed of a link AB with length L:

\[ v = \frac{L}{\frac{1}{n} \sum_{i} t t_{i}} \]  

(3)

where \( n \) is the observation count in a defined interval and \( t t_{i} \) is the travel time of \( i^{th} \) observation.

4.3. Travel Time Prediction

4.3.1. K-Nearest Neighbor (k-NN) method

Travel time at any timestamp is related to the travel time of its close temporal proximity. Therefore, DTT is modeled by the nearest ATT.

\[ D T T_{t} = A T T_{t} + \Delta t t_{t} \]  

(4)

where \( \Delta t t_{t} \) = predicted difference of ATT and DTT at a time-step \( t \).

Travel time difference, \( \Delta t t_{t} \) is predicted by the distance weighted k-NN method using historic daily data:

\[ \Delta t t_{t} = \frac{\sum_{d} w_{d,t} \Delta t t_{d,t}}{\sum_{d} w_{d,t}} \]  

(5)
where $\Delta t_{d,t}$ is the difference between ATT and DTT on a historic day, $d$ at a time-step, $t$ and $w_{d,t}$ is the corresponding weight which is the measure of the similarity between two traffic patterns: the traffic pattern of the present day and of the historic day, $d$. This similarity is the reciprocal of the measure of the variation between those two traffic patterns. This variation is measured by the Euclidian squared distance of two $n$-dimensional vectors representing the latest travel times (ATT of $n$-steps) from the present day, $p$ and historic day, $d$:

$[ATT_t, ATT_{t-1}, ATT_{t-2} \ldots \ldots ATT_{t-n+1}]_p$ and $[ATT_t, ATT_{t-1}, ATT_{t-2} \ldots \ldots ATT_{t-n+1}]_d$.

Since, the $n$-steps should be the steps that have the most significant impact on the current step $t$ to reflect the traffic pattern, the determination of $n$ is a heuristic approach. In this study, travel time was predicted for $n = 1, 2, 3, \ldots, 10$.

4.3.2. Boosting: Least Square Regression (LSBoost) method

LSBoost is a least square regression boost approach that fits regression ensembles to minimize mean-squared error. At each step, a new learner is fitted to the difference between the observed response and the aggregated prediction of all learners grown previously. The following figure represents the algorithm:

Initialize $F_0(x) = \bar{y}$

for $m = 1$ to $M$ do:

$\bar{y}_i = y_i - F_{m-1}(x_i), \ i = 1, N$

$(\rho_m, \alpha_m) = \arg\min_{\rho, \alpha} \sum_{i=1}^{N} [\bar{y}_i - \rho h(x_i; \alpha)]^2$

$F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$

end for

Output the final regression function $F_m(x)$.

Fig. 3. The least square regression boost algorithm (Friedman, 2001).

4.3.3. Kalman Filter method

ATT is chosen for observation, as it is the observation nearest the DTT to be predicted. After rearranging Equation 4 as $ATT_t = DTT_t - \Delta t_{t}$ and treating $-\Delta t_{t}$ as the observation noise $(v_t)$, observation equation shows a linear relationship between ATT and DTT in Equation 6:

$ATT_t = DTT_t + v_t$  \hspace{1cm} (6)

where, $v$ denotes the observation noise. This equation is equivalent to a standard KF observation equation such as $z_t = Hx_t + v_t$ where $z$ and $x$ represents observation and state variables (or ATT and DTT in this case) respectively. The observation matrix, $H$ is assumed to be 1.

The current traffic condition is more correlated with close conditions than it is with distant conditions. The proposed state-space equation is:

$DTT_t = \Phi_{t-1}DTT_{t-1} + w_t$  \hspace{1cm} (7)

And the transition function, $\Phi_{t-1}$:
where $w$ denotes the state-space noise. The proposed state-space equation is similar to a standard KF state-space model, $x_t = \Phi_{t-1}x_{t-1} + w_t$ where $x$, the state variable, is replaced with the DTT.

The proposed KF model based on (Welch and Bishop, 2006) is described below:

**Initial estimates for $\widehat{DTT}_0$ and $P_0$**

**Fig. 4.** KF model.

The most recent DTT ($t - 1$ and $t - 2$ steps) is unavailable until the vehicles have finished travelling the route; therefore, the DTT from the latest and same historical day of week at $t - 1$ and $t - 2$ steps are used to estimate $\Phi_t$. $w$ and $v$ are assumed to be independent of each other and follow the normal probability distributions: $p(w) \sim N(0, Q)$ and $p(v) \sim N(0, R)$.

A priori ($\widehat{DTT}_t$), according to its definition, should be equal to a corresponding DTT. Therefore, the differences between a priori (i.e. state-space projection) and DTT are considered the state-space noise, the State-space noise ($w_t$) is measured by:

$$w_t = \widehat{DTT}_t - DTT_t$$

where $\widehat{DTT}_t$ = Corresponding a priori of DTT at time $t$.

The observation - after the noise is removed - should be equal to the predicted DTT or a posterior, according to eq. (6). Therefore, the difference between observation and predicted time (i.e. DTT-ATT) is considered as the observation noise. The observation noise ($v_t$) at time $t$ can be expressed as:

$$v_t = DTT_t - ATT_t$$

Considering computational simplicity and the availability of sufficient data, noise from historic all-days (instead of same-days) was used to estimate the noise covariance. Assumptions regarding the temporal characteristics of noise can be categorized into three types:
a) Steady Noise (SN): Regardless of time of day, noise is assumed to be the same for a day and estimated from the complete training dataset.

b) Contextual Noise (CN): Noise is assumed to vary by traffic state (free-flow, delay, recurrent, and non-recurrent congestions). Therefore, the entire dataset is divided into four subsets – free-flow, delay, recurrent, and non-recurrent congestions – based on the travel time. Four covariance matrices \( Q_{ff}, Q_{dl}, Q_{rc} \) and \( Q_{nrc} \) are estimated using these sub-datasets.

c) Time-varying Noise (TVN): Noise is assumed to vary with every time step of prediction; hence, the covariance matrix \( Q_t \) is estimated by the noise of a training dataset at time \( t \). In other words, covariance matrices are estimated by splitting the entire dataset into 1,440 subsets for 1,440 intervals of a day. Therefore, \( Q = \{q_1, q_2, ..., q_{1440}\} \) depending on \( t^{th} \) interval of a day.

4.4. Prediction Performance Evaluation Criteria

The mean absolute error (MAE), mean absolute percentage error (MAPE) and root-mean-square error (RMSE) were applied to evaluate the performance of travel time prediction methods. In general, lower value of the MAE, MAPE and RMSE indicates the superiority in prediction. However, a lower MAE, MAPE and RMSE for a corridor that experiences free-flow condition, not necessarily prove that the prediction method is superior. Since travel time of such a corridor mostly remains unchanged for a short time interval, a naïve method (e.g. using current travel time as the 5-min ahead prediction), as opposed to an advanced method, might work best. Therefore, it is important to quantify the performance of the naïve method and compare the results with the results of advance methods for better understanding. The comparison helps to quantify the overall improvements made by an advance method. In this study, the MAE, MAPE and RMSE from the prediction results of a naïve method, considering that using ATT as the prediction of DTT is a naïve method, was termed as Actual MAE, Actual MAPE and Actual RMSE (i.e. AMAE, AMAPE and ARMSE) respectively.

5. Results & discussion

The complete dataset in this study was divided into two sets: Training and Validation. Twenty-eight of forty-seven days’ worth of data was used for the training dataset, and the rest of the data was used as the validation dataset. It was more appropriate to use link speed data as opposed to link travel time due to the variability in link lengths. The prediction performance index was utilized to perceive global performance (i.e. the performance of the entire network). The MAPE was calculated based on the travel time dataset in order to discern the local performance at each link. Quantifying improvement is impossible without knowing the AMAPE or actual lag/gap. Considering that using ATT as the prediction of DTT is a naïve method, AMAPE (i.e. MAPE of ATT) was used as the benchmark for MAPEs generated by other methods.

The k-NN method showed different prediction performance of different values of \( n (n = 1, 2, ..., 10) \). Fig. 5 represents the local (i.e. each link) performance of the k-NN model for validation dataset when \( n = 1, 3 \) and 5. When evaluating criteria of local performance, the MAPE of the prediction for each link should be smaller than that of the actual lag/gap (AMAPE or the benchmark).
In Fig. 5, MAPEs of travel times predicted by the k-NN method using \( n = 1, 3 \) and 5 are compared to the actual lag/gap. The local performance is unacceptable since the MAPEs of more than half of the links are greater than the actual lag/gap. Performance shows a decreasing trend (i.e. MAPE increases) with the increase of \( n \). Moreover, the global performance measured by the MAPE over entire route is poor. The overall MAPEs are higher than the actual gap or AMAPE (6.70%), suggesting the lack of repeating traffic condition over the entire period. Therefore, k-NN is not an appropriate method to predict DTT from ATT when the traffic conditions in the training dataset are not similar to the conditions in the validation dataset.

Two new variables, day of the week and time of the day, were included besides ATT in the LSBoost application. In addition, the latest ATTs (2, 3, 5, 10 minutes etc.) were combined with the current ATT to create a new variable. Similar prediction performances were observed for different numbers of newly added ATT columns (\( n = 2, 3, 5 \) and 10). In other words, this algorithm shows limited sensitivity to the size of the combined ATTs. For visual clarity, only two selected results with the actual gap are shown in Fig. 6 below:
In Fig. 6, MAPEs of travel times predicted by LSBoost using two, three and five previous-step ATTs (n = 2, 3 and 5 respectively) with current ATT are compared to the actual lag/gap. The MAPEs and AMAPEs of different links on I-90 have very similar values which indicate the shortcoming of this method on a freeway that experiences free-flow. The performance of LSBoost prediction is notably well for the Beltline Highway that experiences congestion. The link (22nd) that connects I-90 and Beltline corridor performs very poorly showing MAPE much higher than that of AMAPE. Therefore, LSBoost is not recommended at this point to predict DTT from ATT for freeways.

Spatial noise characteristics for KF models were assumed in this study by considering that a) noise of different links in a particular corridor can have similar characteristics, and b) noise of different links, regardless of corridor, can have different characteristics. Previously, three categorical assumptions regarding temporal characteristics of noise have been discussed; therefore, the output of the KF model would be affected by six different estimation procedures of noise covariance regarding the spatial-temporal characteristics of noise. The six methods are: corridor-based steady noise (CB-SN), contextual noise (CB-CN), time-varying noise (CB-TVN), link-based steady noise (LB-SN), contextual noise (LB-CN), and time-varying noise (LB-TVN).

Appropriate noise characteristics of KF should be determined through the evaluation of local performance. Fig. 7, Fig. 8, and Fig. 9 represent the local (i.e. each link) performance of the KF model for validation dataset with different noise assumptions. When evaluating criteria of local performance, the MAPE of the prediction for each link should be smaller than that of the actual lag/gap (AMAPE or the benchmark). In Fig. 7, MAPEs of prediction by the KF model using CB-SN and LB-SN are compared to the actual lag/gap. The prediction performances of the KF model using both CB-SN and LB-SN for each individual link are acceptable, as no link shows a MAPE greater than the actual lag/gap.
Fig. 7. MAPE of KF at each link for steady noise assumption vs. actual gap.

In Fig. 8, MAPEs of travel times predicted by the KF model using CB-CN and LB-CN are compared to the actual lag/gap. It is clear that the KF model with both CB-CN and LB-CN has a few links’ MAPE greater than AMAPE. In general, the context-based noise assumption is supposed to perform better for the corridor that experiences congestion. The poor performance could be due to the stability of travel time resulting from the saturated traffic flow rate under congestions. At a saturated flow, the variations between ATT and DTT become similar to the variations in free flow or delay conditions.

Fig. 8. MAPE of KF at each link for contextual noise assumption vs. actual gap.
In Fig. 9, MAPEs of travel times predicted by the KF model using CB-TVN and LB-TVN are compared to the actual lag/gap. The assumption of CB-TVN is invalid for a corridor with some links that experience traffic congestion. Despite the improvement from contextual noise assumption, Fig. 9 shows that the noise homogeneity assumption of CB-TVN is violated at Link 28, as its performance exceeds the actual lag/gap (AMAPE).

![Figure 9](image-url)  
**Fig. 9.** MAPE of KF at each link for time varying noise assumption vs. actual gap.

The above discussion provides a comprehensive description of local/link performance of different noise assumptions in a KF model. The unambiguous analyses reflect the suitability of CB-SN, LB-SN and LB-TVN. Table 1 shows the global performance of prediction expressed by the MAE and RMSE of speed data calculated from travel time, which reaffirms the most appropriate noise assumption for the dataset is LB-SN.

**Table 1**

<table>
<thead>
<tr>
<th>Noise Assumption</th>
<th>Training Dataset</th>
<th>Validation Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>RMSE</td>
</tr>
<tr>
<td>CB-SN</td>
<td>2.25</td>
<td>6.31</td>
</tr>
<tr>
<td>LB-SN</td>
<td>2.17</td>
<td>5.96</td>
</tr>
<tr>
<td>LB-TVN</td>
<td>2.15</td>
<td>6.78</td>
</tr>
</tbody>
</table>

The KF model with LB-SN assumption is more accurate and computational efficient. For instance, the run time of KF with corridor-based noise assumptions is approximately 15mins, whereas link-based assumptions take only a minute or two. Corridor-based noise homogeneity assumption calls for extra processing of data since Bluetooth data is collected over each link; this way, noise covariance can be estimated over the entire corridor. This extra processing increases
the run time significantly. The link-based model is the most suitable for an on-line application.
KF with LB-SN was selected to predict DTT from ATT.

The MAPEs of the training and validation datasets using LB-SN KF are 4.27% and 4.53%, respectively, whereas, the actual lags/gaps (i.e. AMAPEs) are 6.43% and 6.70%, respectively, based on travel time dataset. Improvements are significant when the smaller actual lag/gap is considered, including a 50% reduction in AMAPE in some links. When the KF model with LB-SN assumption is applied, the validation dataset shows that a 40-50% gap between ATT and DTT is minimized in different links of the I-90 corridor, and a nearly 30-40% gap is minimized in different links of the Beltline Highway corridor.

Moreover, the KF model with the LB-SN assumption has a superior prediction performance in cases of traffic state transition (e.g. onset and end of congestion). Fig. 10(a) represents the DTT, ATT, and predicted travel time, and Fig. 10(b) demonstrates the actual lag/gap and prediction error corresponding to Fig. 10(a).

**Fig. 10.** Prediction performance between free flow and congested conditions.

Fig. 10 clearly depicts that the ATT, DTT, and predicted travel time are almost equal at free flow conditions (before 16:30:00). Actual and prediction errors are negligible after 18:15:00 when congestion is stable. Fig. 10(a) Box A shows the onset of congestion where the actual lag/gap is more than 50s, and Fig. 10(b) shows prediction lag is less than 20s. Box B in Fig. 10(a) shows the end of the congestion situation where the actual and prediction lags show little difference (less than 20s). However, at 17:15:00, the mid-point of the end of congestion, the ATT is off by 100s from DTT while the prediction error is around 30s. Despite of having a moderate improvement (around 40%) according to MAPE (global performance), the prediction
shows excellent improvement (around 70%) in cases such as that in box B where there is a state transition. Since such cases cover shorter time periods compared to the complete study period, the overall performance (local or global) indices are unable to represent the robustness of the prediction algorithm. Therefore, the selected noise assumption based on the KF model is capable of predicting travel time from the travelers’ point of interest (the onset and end of congestion rather than free-flow or stable congested conditions).

To conclude, to predict the current traffic state as well as travel time, k-NN and LSBoost utilize the full dataset whereas the KF method uses most recent dataset. Intrinsically, the heteroscedasticity of traffic condition captured by the most recent data is more resemble to the current. Perhaps, this gives KF advantages over other methods. Moreover, the noise modeling capability offers additional benefits such as the selection of a subset of data to represent the outstanding heteroscedasticity that may not be adequately represented by the most recent data used in the state projection. Hence, prediction performances of the KF method with noise assumptions are more consistent than that of the k-NN and LSBoost in context of different roadway links.

6. Conclusion

ATT (or $t_{expm}$) is the most available form of travel time, but DTT (or $t_{expr}$) is the most desirable. The k-NN, LSBoost and KF algorithms were used to predict DTT and thus assist motorists by providing a more accurate and reliable travel time. Overall, KF outperformed other two methods. Although ATT and DTT differ slightly when the flow of traffic is stable, the variation becomes significant when the traffic state is in transition (e.g., moving from unstable to stable or vice versa). The KF algorithm with steady noise assumption captured a state of transition property accurately and provided an excellent prediction. KF is fast for link-based applications, making it desirable for data sources that contain route travel time split into shorter links (e.g., loop detectors/Bluetooth data). Although KF is applied (by default) to each link that is isolated as a different model, it demonstrates a higher level of accuracy and faster speed due to its flexibility, simplicity, and compatibility with data characteristics. The application was demonstrated during peak periods on freeways covering two corridors—one with fewer transitions in traffic state and the other with frequent transitions. Diversity in noise assumptions showed negligible impact on the former, while steady noise assumption showed better performance on the latter. Steady noise (SN) refers to a fixed covariance estimated from the complete training dataset, which indicates that the KF model performs better with the generalized noise assumption. Hence, the KF with LB-SN assumption was preferred over other assumptions. Better performance was observed during a time when transitions in traffic state occurred more frequently.

Since arterial highways are supposed to exhibit more state transitions, future research should examine the performance of predicting arterial highway travel time to test the robustness of this method. Furthermore, as firstly pointed out in Guo et al. 2012, vehicular traffic condition series is heteroscedastic in nature, new research is needed to investigate the second-order moment of travel time, i.e., the uncertainty of the time series data of travel time. Future research can
substantiate and expand on the pioneering research methodologies in the travel time uncertainty domain (Guo et al. (2014), Huang et al. (2017), Guo et al. (2017) and Cao et al., (2017)).

References


Toppen, A. & Wunderlich, K. 2003. Travel time data collection for measurement of advanced traveler information systems accuracy, Mitretek Systems.


Welch, G. & Bishop, G. 2006. An introduction to the Kalman filter. Department of Computer Science, University of North Carolina, Chapel Hill, NC, unpublished manuscript.


