Travel Time Prediction for Congested Freeways With a Dynamic Linear Model

Semin Kwak⑩, *Member, IEEE*, and Nikolas Geroliminis⑩, *Member, IEEE*

Abstract—Accurate prediction of travel time is an essential feature to support Intelligent Transportation Systems (ITS). The non-linearity of traffic states, however, makes this prediction a challenging task. Here we propose to use dynamic linear models (DLMs) to approximate the non-linear traffic states. Unlike a static linear regression model, the DLMs assume that their parameters are changing across time. We design a DLM with model parameters defined at each time unit to describe the spatio-temporal characteristics of time-series traffic data. Based on our DLM and its model parameters analytically trained using historical data, we suggest an optimal linear predictor in the minimum mean square error (MMSE) sense. We compare our prediction accuracy of travel time for freeways in California (I210-E and I5-S) under highly congested traffic conditions with those of other methods: the instantaneous travel time, k-nearest neighbor, support vector regression, and artificial neural network. We show significant improvements in the accuracy, especially for short-term prediction.

Index Terms—Travel time prediction, dynamic linear model, time-varying coefficients, least-squares, minimum mean square error.

I. Introduction

TRAVEL time prediction is one of the essential features to support successful Intelligent Transportation Systems (ITS). An accurate prediction of travel time not only helps travelers to make decisions about their trips but also enables traffic operators to develop successful control strategies. This necessity has engaged many researchers on the topic of travel time forecasting despite the vast amount of existing literature.

The methods for predicting travel time can be categorized into model-based and data-driven approaches [1]. The model-based methods predict future traffic parameters (e.g., occupancy, flow, or speed) by building a traffic model, such as the Cell Transmission Model [2], [3], the queuing theory [2], [4], [5], or macroscopic traffic flow model [6]. These model-based methods provide a straightforward interpretation of the predicted results because of their physical intuition, such as flow dynamics.

The data-driven methods, on the other hand, predict travel time by extracting specific features from traffic data. Common data-driven methods include Linear Regression [7], [8], Autoregressive models [9]–[12], Kalman Filtering [13]–[16] and Bayesian inference [17], [18]. These methods predict travel time by assuming that all the data satisfies a certain probabilistic distribution.

Furthermore, the increased accessibility to traffic data and the improved computing power in these days allow researchers to develop more sophisticated data-driven algorithms, such as Support vector regression (SVR) [19]–[21], Artificial neural networks (ANN) [22]–[29], Long Short-Term Memory Network [30], [31] and Ensemble learning [32]–[34].

Conversely, the travel time predictors can also be categorized into *direct* and *indirect* methods. Direct methods contain a straightforward approach to minimize the error in predicted travel time [7]–[12], [14]–[25], [29], [30], [33]. The main advantage of the direct methods lies in their simplicity since they take into account only the travel time as an output. However, the prediction performance can also be degraded as all the complex traffic characteristics are assumed to be reflected in travel time. Another limitation of these methods is that they require separate models for different circumstances; for example, when the departure time or the origin location change a new model has to be trained for that exact setting.

In contrast, the indirect methods estimate travel time by predicting future traffics first, such as velocity or occupancy field. Then they use the predicted traffic states to estimate travel time [13], [32], [35], [36]. By definition, the predicted traffic states can also be re-used to predict those for the next horizon. Contrarily to the direct methods, predicting future traffic parameters allows the model to estimate a travel time for any scheduled departure time or space, which makes the indirect method more versatile.

In this paper, we suggest a method based on a dynamic linear model to predict the velocity field and therefore travel time, which falls into the intersection of the indirect method and the data-driven approach. Using historical data, we analytically find the model parameters in the least-squares sense. We compare the proposed method with four other predictors that are used in the literature: the instantaneous travel time, the k-nearest neighbor [35], artificial neural networks [22], and the support vector regression [19]. Our comparison shows that the proposed method has a great potential to improve the short-term prediction accuracy as well as to become a versatile tool in various traffic situations with this stand-alone model.

II. METHOD

A. Dynamic Linear Model

We suggest a dynamic linear model for speed and travel time prediction. The dynamic characteristics allow the model to extract temporal features of the parameters of interest (velocity fields in our case). The model describes a linear relationship between velocities at a specific time t_k and the next step time t_{k+1} using the following equation:

$$\mathbf{v}_{k+1}^d = H_k \mathbf{v}_k^d + \mathbf{n}_k^d, \quad \forall d, \ \forall k \in \{0, \dots, K-1\}. \quad (1)$$

Here the vector $\mathbf{v}_k^d$ refers to a velocity vector at time t_k on day d, which can be expressed as follows:

$$\mathbf{v}_k^d = \begin{bmatrix} v^d(x_1, t_k) \\ \vdots \\ v^d(x_M, t_k) \end{bmatrix} \in \mathbb{R}^{M \times 1}, \quad (2)$$

where the constant M represents the number of measured velocities on different locations on a freeway of interest, for example with data from loop detectors. We define a velocity field as a scalar function of time t and position x:

$$v^d(x, t) \in \mathbb{R}, \quad (3)$$

where each point of the velocity field represents a measured velocity value.

In Eq. (1), the second vector $\mathbf{n}_k^d$ on the right-hand side refers to a noise vector which we assume to follow a Gaussian distribution with a zero mean and a variance σ^2 under independent and identically distributed (i.i.d.) conditions, i.e.:

$$\mathbf{n}_k^d \sim \mathcal{N}\left(\mathbf{0}_M, \sigma^2 I_M\right), \quad \forall k \in \{0, \dots, K-1\}, \quad (4)$$

where $\mathbf{0}_M$ and I_M are the vectors with all zero entities and the identity matrix of size M, respectively.

Matrix H_k in Eq. (1) is a transition matrix, which represents a linear relationship between the two velocity vectors $\mathbf{v}_k^d$ and $\mathbf{v}_{k+1}^d$ according to the time t_k and t_{k+1}. In particular, the diagonal elements of H_k describe a direct temporal relationship at each specific location, whereas the off-diagonal terms of H_k contain the spatio-temporal relationship between two consecutive velocity fields. The first aim of this paper is to find an analytical solution of the transition matrix H_k for every possible time t_k so that we build a dynamic transition matrix.

The model presented in Eq. (1) suggests three important factors. First, the velocity vector $\mathbf{v}_k^d$ is *linearly* transformed to the vector $\mathbf{v}_{k+1}^d$ with an additive Gaussian noise. The linearity and the Gaussian noise assumption allow the transition matrices H_k for every k to be trained analytically, which will be discussed in the next section. Secondly, the transition matrix is defined at each time unit such that the model captures a non-linear traffic flow over time even though it is based on a linear regression model for a given period. Lastly, the transformation matrix of two consecutive time steps k and $k+1$ is set regardless of different traffic profiles, which means that the matrix H_k does not depend on, for example, the days of a week. We will show later that this framework captures well traffic conditions of different days.

B. Estimation of Model Parameters

We shall estimate the transition matrices H_k for all k values with historical data set using the least-squares method. Within a set of days $\mathbb{D}$ we choose for estimation (or we call it a day set), Eq. (1) can be extended as

$$V_{k+1}^{\mathbb{D}} = H_k V_k^{\mathbb{D}} + N_k^{\mathbb{D}}, \quad (5)$$

where the matrix $V_k^{\mathbb{D}}$ is a time-velocity matrix defined for the day set $\mathbb{D}$ for a specific time t_k as a collection of all velocity vectors corresponding to the same time index within $\mathbb{D}$, i.e.:

$$V_k^{\mathbb{D}} = \begin{bmatrix} \mathbf{v}_k^{d_1} & \mathbf{v}_k^{d_2} & \dots & \mathbf{v}_k^{d_{|\mathbb{D}|}} \end{bmatrix} \in \mathbb{R}^{M \times |\mathbb{D}|}, \quad \forall d_i \in \mathbb{D}. \quad (6)$$

Here, the operator $|\cdot|$ of a set represents the cardinality (the number of elements) of the set. Therefore, the number of the rows and columns represents the data dimension and the size of a day set, respectively.

From Eq. (5), we shall estimate the transition matrix using the least-squares method, which is also equivalent to the solution of the maximum likelihood method since we assume i.i.d. Gaussian noise [37]. Therefore, the optimization problem can be stated as:

$$\underset{H_k}{\text{minimize}} \left\| V_{k+1}^{\mathbb{D}} - H_k V_k^{\mathbb{D}} \right\|_F^2, \quad (7)$$

where the operator $\|A\|_F = \sqrt{\text{tr}\left(A A^\top\right)}$ and $\text{tr}\left(A A^\top\right)$ indicates a sum of the all diagonal elements of a matrix $A A^\top$.

In order to prevent an ill-posed problem and to give priority to more recent data for better prediction, we introduce an adaptive matrix regularization term with a regularization parameter ρ and a forgetting factor λ, which is recursively multiplied to old data set, to Eq. (7) as follows:

$$\underset{H_k}{\text{minimize}} \ \rho \lambda^{|\mathbb{D}|} \|H_k\|_F^2 + \left\| \left(V_{k+1}^{\mathbb{D}} - H_k V_k^{\mathbb{D}} \right) \Lambda_{|\mathbb{D}|}^{\frac{1}{2}} \right\|_F^2, \quad (8)$$

where the diagonal matrix Λ_N is defined as follows with the forgetting factor λ:

$$\Lambda_N = \begin{bmatrix} \lambda^{N-1} & 0 & \cdots & 0 \\ 0 & \lambda^{N-2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}. \quad (9)$$

The first term in Eq. (8) is the regularization term; its major role is to prevent the transition matrices from overfitting to a small training data set. The term also allows reliable estimation of the transition matrix numerically, which is described in Appendix A-B.

The forgetting factor, on the other hand, decreases the weight of old data exponentially during the recursive training process. For instance, when $\lambda = 0.995$, a set of data a year ago is penalized by the factor of $(0.995)^{365} = 0.16$. This forgetting factor also allows the regularization term to vanish, adapting to the size of the training set since the term converges to zero when the number of elements in $\mathbb{D}$ is getting bigger. The modified problem in Eq. (8) will be equivalent to the original problem of Eq. (7) when we set $\rho = 0$ and $\lambda = 1$, which means no regularization and no forgetting process.
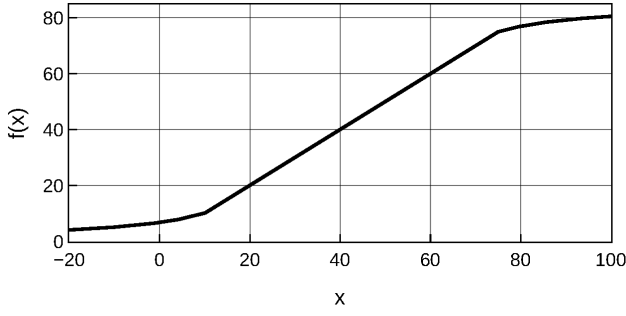
Fig. 1. Post-processing function for the freeways studied in this paper. The function makes the output values fall within a reasonable speed range, i.e., $0 \leq f(x) \leq 85$.

The optimization problem in Eq. (8) can be analytically solved, and we derive it in Appendix A-A. Here we present the solution:

$$\bar{H}_k^{\mathbb{D}} = V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}}\right)^{\top} \left(V_t^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}}\right)^{\top} + \rho \lambda^{|\mathbb{D}|} I_M\right)^{-1},$$
(10)

where the notation $\bar{A}$ represents an estimator of A.

One popular property of the solutions of the least squares problem is that they can be updated with new observations [37]. The updating method not only prevents increasing memory size, but also makes computation time consistent since the procedure only needs a pre-trained model and a new observation for an update. This property can be essential to support accurate travel time prediction because the system should be up-to-date with time. We describe the implementation of an updating algorithm for Eq. (10) in Appendix A-C.

C. Velocity Prediction

With the transition matrices $\bar{H}_k^{\mathbb{D}}$ introduced in the previous section, we can now predict the velocity vectors for traffic forecasting, which is the second aim of this paper. We start by setting a notation of a predictor for i-step ahead at time step k:

$$\mathbf{v}_{k+i|k}^{\tilde{d}} \quad \text{for } i = 1, 2, \cdots \text{ and } \tilde{d} \notin \mathbb{D}.$$
(11)

Assuming that the trained transition matrix is close enough to the truth i.e., $\bar{H}_k^{\mathbb{D}} \approx H_k$ for all k, the velocity vector $\mathbf{v}_{k+i}^{\tilde{d}}$ is written as follows using Eq. (1):

$$\mathbf{v}_{k+i}^{\tilde{d}} = \bar{H}_{k+i-1}^{\mathbb{D}} \mathbf{v}_{k+i-1}^{\tilde{d}} + \mathbf{n}_{k+i-1}^{\tilde{d}}$$
(12)

$$= \bar{H}_{k+i-1}^{\mathbb{D}} \left(\bar{H}_{k+i-2}^{\mathbb{D}} \mathbf{v}_{k+i-2}^{\tilde{d}} + \mathbf{n}_{k+i-2}^{\tilde{d}}\right) + \mathbf{n}_{k+i-1}^{\tilde{d}}$$
(13)

$$\vdots$$

$$= \bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}} + \mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}},$$
(14)

where

$$\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} = \prod_{j=1}^{i} \bar{H}_{k+i-j}^{\mathbb{D}},$$
(15)

$$\mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} = \sum_{j=1}^{i-1} \bar{H}_{k+i-j}^{\mathbb{D}} \mathbf{n}_{k+i-j-1}^{\tilde{d}} + \mathbf{n}_{k+i-1}^{\tilde{d}}.$$
(16)

The noise vector in Eq. (14) follows a zero mean Gaussain vector with a covariance Σ since a linear combination of zero mean Gaussian random variable follows another zero mean Gaussian random variable [38]. As a result,

$$\mathbf{v}_{k+i}^{\tilde{d}} \sim \mathcal{N}\left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}}, \Sigma\right).$$
(17)

We choose a predictor as the maximizer of the above density function:

$$\mathbf{v}_{k+i|k}^{\tilde{d}} = \bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}}.$$
(18)

This predictor is also an optimal estimator of the linear minimum mean square error (LMMSE) (Appendix A-D). Therefore, Eq. (18) shows that the best linear predictor $\mathbf{v}_{k+i|k}^{\tilde{d}}$ is the propagation of the current measurement $\mathbf{v}_k^{\tilde{d}}$ through the trained transition matrices from $\bar{H}_k^{\mathbb{D}}$ to $\bar{H}_{k+i-1}^{\mathbb{D}}$.

However, in rare cases, an unbounded solution $\mathbf{v}_{k+i|k}^{\tilde{d}}$ can have a negative speed or an unrealistically high speed due to the Gaussian noise assumption. This kind of wrong estimations severely distort the calculation of travel time. In order to correct this effect, we design a post-processing function $f(x)$:

$$f(x) = \begin{cases} b \cdot \dfrac{a(x - \tau_l)}{1 + |a(x - \tau_l)|} + \tau_l & x < \tau_l \\ x & \tau_l \leq x \leq \tau_u \\ b \cdot \dfrac{a(x - \tau_u)}{1 + |a(x - \tau_u)|} + \tau_u & x > \tau_u, \end{cases}$$
(19)

where the constants a and b are smoothing parameters, and τ_l and τ_u are threshold parameters. We empirically set the smoothing parameters a and b to be 0.05 and 10, respectively. We also have empirically chosen the lower threshold value τ_l and the upper threshold value τ_u as 10 and 75 (mph), respectively.

Figure 1 shows the post-processing function with the chosen parameter sets. The input x in this example is a velocity value. When x is below the lower threshold of 10, the function deflects the values to be always positive. When x is above the upper threshold of 75, the function makes the output converging to the upper limit, which is 85 miles per hour in our case. Between the two boundaries, it does not change the input value. We have tested different sets of the smoothing and threshold parameters and found that it has little impact on prediction results.

We apply this post-processing function in Eq. (19) to Eq. (18) recursively at each step of multiplication so that we can exclude the invalid estimations. The following shows the detailed procedure.
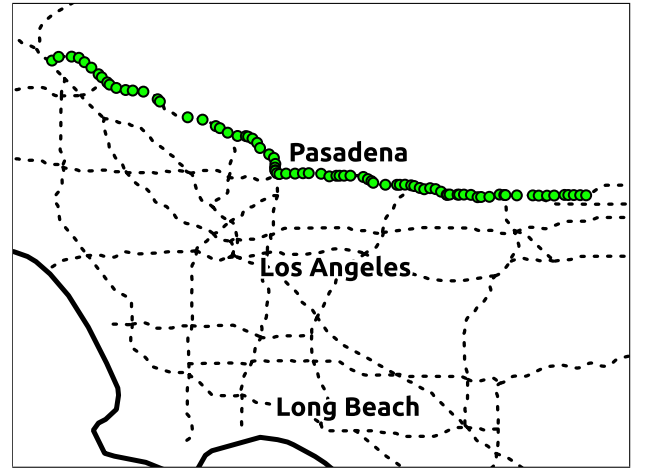
$$\mathbf{v}_{k+1|k} = f\left(\bar{H}_k \mathbf{v}_k\right)$$

$$\mathbf{v}_{k+2|k} = f\left(\bar{H}_{k+1} \mathbf{v}_{k+1|k}\right)$$

$$\vdots$$

$$\mathbf{v}_{k+i|k} = f\left(\bar{H}_{k+i-1} \mathbf{v}_{k+i-1|k}\right)$$
(20)

D. Travel Time Estimation

For estimating the travel time of a moving vehicle, we assume that the vehicle experiences a velocity field, which
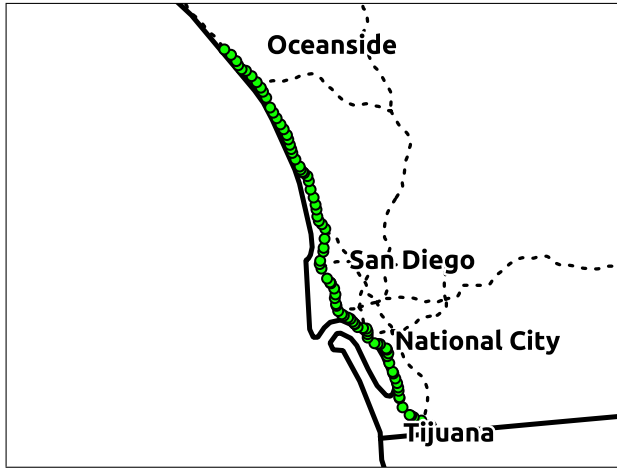
Fig. 2. Detector locations on the freeways I5-S and I210-E. The 88 loop detectors along the freeway I5-S and the 83 loop detectors along the freeway I210-E are used in this paper (green dots on both figures). In the names of the freeways, S (south) and E (east) represent the direction of the freeways.

is a function of time t and space x, and we know the exact continuous velocity field $v(t, x)$. Then we can calculate the increment of time Δt after traveling a distance Δx as

$$\Delta t = \frac{1}{v(t, x)} \Delta x, \tag{21}$$

since $v = dx/dt$. Consequently, a travel time at time t_0 given a velocity field $v(t, x)$ is computed recursively as follows:

Algorithm 1 Numerical Calculation of Travel Time

Input: the velocity field $v(t, x)$; the departure time and location, t_0 and x_0; the location of the destination x_M; and the space increment Δx
Output: the travel time
1: **Initialization:** $t \leftarrow t_0$, $x \leftarrow x_0$
2: **while** $x < x_M$ **do**
3:     $t \leftarrow t + \frac{1}{v(t,x)} \Delta x$
4:     $x \leftarrow x + \Delta x$
      return $t - t_0$

In reality, we only know a discretized velocity field instead of a continuous one. For our study, we know velocities at each sensor (every 0.7 miles on average) every 5 minutes. We generate the continuous velocity field by interpolating the discretized velocity field with linear bivariate B-spline curve fitting.

E. Performance Measures for Comparison With Other Methods

To measure the performance of our prediction, we use the absolute percentage error (APE) and the mean absolute percentage error (MAPE), which are defined as:

$$\text{APE}(t) = 100 \cdot \left| \frac{a(t) - p(t)}{a(t)} \right|, \tag{22}$$

$$\text{MAPE}(\mathbb{T}) = \frac{1}{|\mathbb{T}|} \sum_{t \in \mathbb{T}} \text{APE}(t), \tag{23}$$

where the set $\mathbb{T}$ represents a set of time elements to examine. The values $a(t)$ and $p(t)$ are respectively the actual travel time and the predicted travel time when departed at time t. The MAPE estimates the mean deviation of estimation to the ground truth (i.e., the experienced travel time) in percentage (%) unit.

III. Results and Discussions

In this section, we employ the proposed method to predict traffic flow and thus travel time using real-world data. We examine the performance of the proposed method by comparing predicted travel time with that of other existing predictors.

A. Traffic Data

We use traffic data of two different freeways in California having different traffic profiles: Freeway I5-S and Freeway I210-E. Along the two freeways, there are respectively 88 and 83 loop detectors within the area of our examination (Fig. 2). The total length of the corridor along I5-S is 58.33 miles, and that of I210-E is 52.14 miles. The loop detectors collect measurements (flow and occupancy data) every 30 second, and we use 5 minutes aggregated speed data, which is processed by the Caltrans Performance Measurement System (PeMS).

From PeMS, we extracted one-year traffic data of both freeways (2012 for I5-S and 2015 for I210-E) for experiments. We allocated the first 70% of traffic data (from January 1st to September 12th) as a training set, the next 15% of data as a validation set (from September 13th to November 11th), and the last 15% of data as a test set (from November 12th to December 31st) for all the experiments.

We have considered the traffic data from 6 AM to 9 PM only and divided the data into two groups: a peak period and an off-peak period (Fig. 3). Since the two freeways have very different traffic profiles, we have defined the peak and off-peak periods differently for each freeway. For Freeway I5-S, the peak period is defined as 6 - 10 AM (morning peak)
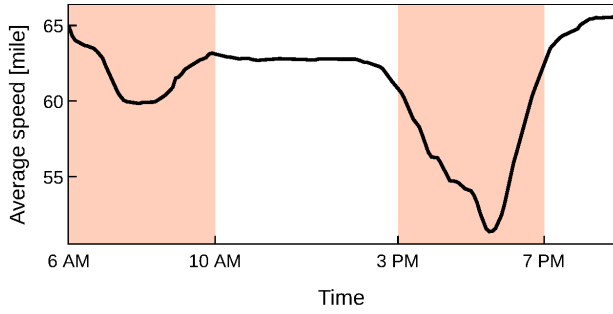
(a) Freeway I5-S (weekdays)
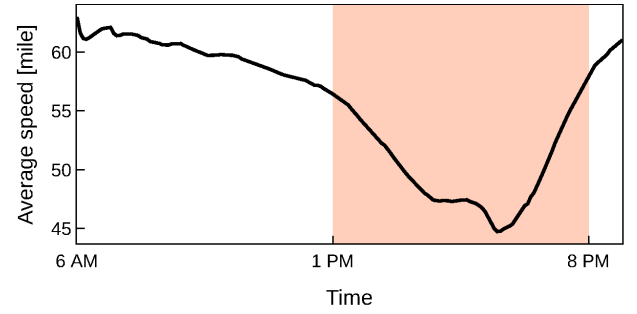
(b) Freeway I210-E (weekdays and Saturday)

Fig. 3. Average speed by time of test set for each freeway. The peak periods are also defined as the area with color.

and 3 - 7 PM (evening peak) on weekdays (from Mondays to Fridays); for Freeway I210-E, it is defined as 1 - 8 PM (afternoon peak) every day except Sundays. The off-peak periods are defined as a complementary set of the corresponding peak periods. Figure 3 illustrates them straightforwardly.

B. Determining Hyper-Parameters

Using Eq. (10), we have trained transition matrices with different pairs of the regularization parameter ρ and the forgetting factor λ. For each freeway, we have trained the transition matrix by all possible combinations of the following sets:

$$\rho \in \{0, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 1000, 3000, 10000\},$$
$$\lambda \in \{1, 0.999, 0.995, 0.99, 0.95\}. \quad (24)$$

Table I shows the MAPE of travel time on the validation sets of the two freeways (peak periods only) by varying the hyper-parameters. The MAPE is not very sensitive to the regularization parameters, but it is influenced by the forgetting factors, as when the value of the forgetting factor is too low, this leads to a training of the transition matrices with not enough data. For each case of the freeways, we have chosen the optimal pair among the tested parameter sets, which are:

$$(\rho, \lambda) = \begin{cases} (3000, 0.995) & \text{for I5-S} \\ (300, 1) & \text{for I210-E.} \end{cases} \quad (25)$$

We show that the optimal pairs of hyper-parameters chosen above work well for traffic prediction by showing an example of predicted velocity fields (Fig. 4). The prediction results (the contour plot) in Fig. 4 (c) and (d) show similar patterns to the ground truths (Fig. 4 (a) and (b)), which confirms that the chosen hyper-parameters are functioning well for predicting speeds and travel time.

C. Comparison of Travel Time With Different Forecasters

We examine the performance of our proposed method by comparing its performance with that of different prediction methods. We have chosen four various forecasters: the instantaneous travel time forecaster (abbreviated to *inst.*) as a real-time measurement-based method; the k-nearest neighbor (k-NN) as a historical data-based method; the support

TABLE I
MEAN ABSOLUTE PERCENTAGE ERROR (MAPE) ON THE VALIDATION SETS FOR FREEWAYS I5-S AND I210-E WITH DIFFERENT HYPER-PARAMETER PAIRS

(a) Freeway I5-S

ρ \ λ	1.000	0.999	0.995	0.990	0.950
0	3.447	3.433	3.414	3.432	5.134
1	3.441	3.428	3.411	3.430	5.134
3	3.430	3.418	3.405	3.427	5.134
10	3.395	3.388	3.385	3.415	5.134
30	3.318	3.319	3.340	3.385	5.134
100	3.183	3.187	3.232	3.317	5.133
300	3.071	3.074	3.106	3.202	5.130
1000	2.996	2.987	2.982	3.045	5.119
3000	3.003	2.987	**2.936**	2.937	5.091
10000	3.244	3.192	3.043	2.926	5.003

(b) Freeway I210-E

ρ \ λ	1.000	0.999	0.995	0.990	0.950
0	4.842	4.879	5.048	5.280	7.456
1	4.858	4.896	5.063	5.290	7.455
3	4.848	4.888	5.058	5.288	7.454
10	4.816	4.859	5.037	5.275	7.453
30	4.747	4.793	4.986	5.239	7.453
100	4.641	4.673	4.865	5.143	7.451
300	**4.602**	4.622	4.729	4.998	7.449
1000	4.643	4.637	4.672	4.808	7.443
3000	4.806	4.781	4.718	4.745	7.427
10000	5.318	5.232	4.965	4.811	7.369

vector regression (SVR) and the vanilla artificial neural network (ANN) as representatives for direct methods. All the details of implementing these methods are explained in Appendix B.

Specifically, we evaluate travel time using these methods with different prediction horizons. We define a travel time at time t with a prediction horizon h-minutes as a travel time that a vehicle will experience when it departs h-minutes after the time t. For example, Fig. 4 (c) and (d) show travel time prediction with a 60-minute horizon at the current time of 2 PM. We assign four different values for h: 0, 15, 30, and 60 minutes.

Figure 5 shows the average prediction errors (APE) of the results on the test sets. It shows that the proposed method always gives the best accuracy among others when $h = 0$
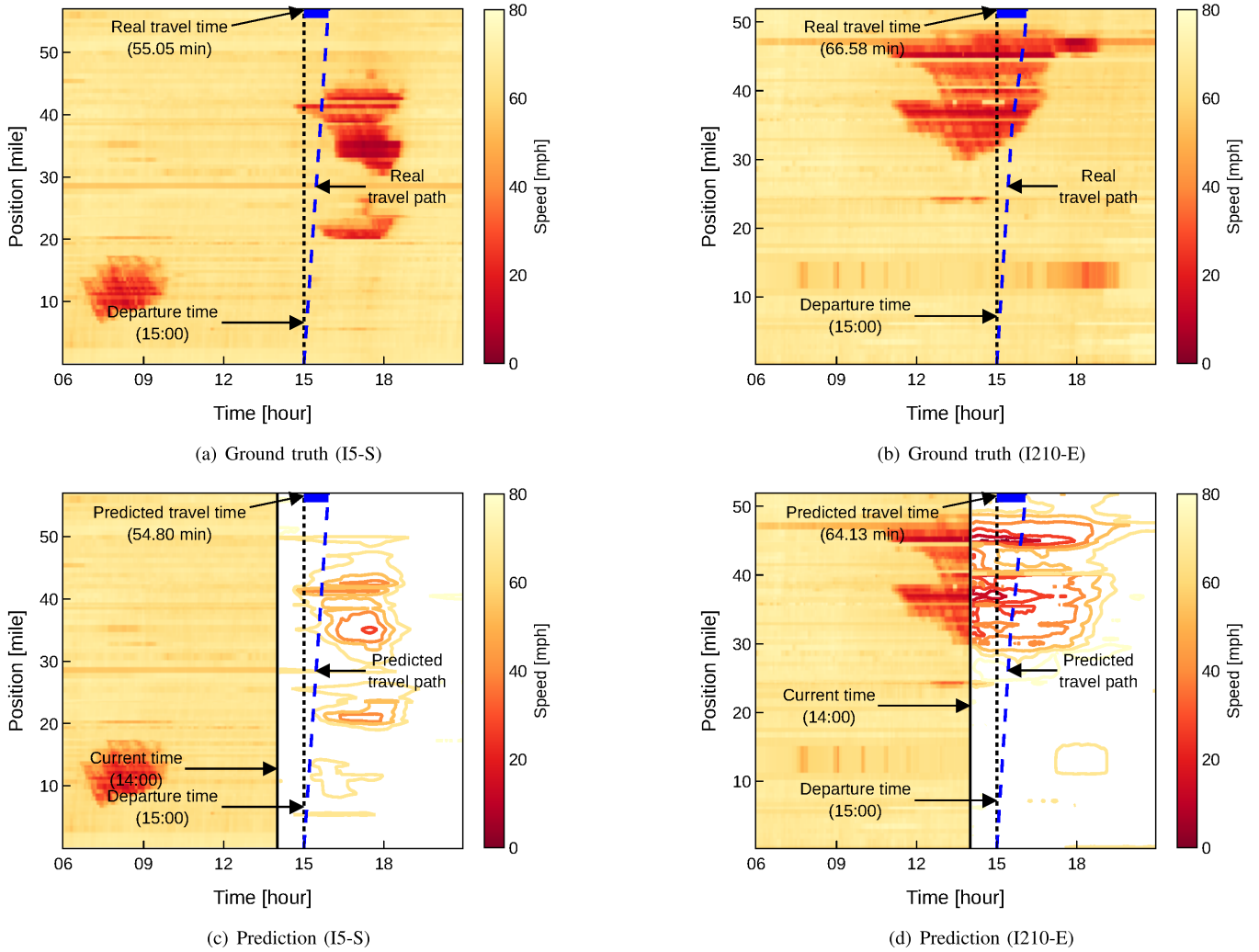
Fig. 4. Velocity field of freeway I5-S on December 4th (Tuesday), 2012 and I210-E on December 25th (Friday), 2015. (a) and (b): The ground truths and (c) and (d): predicted velocity fields at 2 PM and afterwards using the proposed method, which is represented as the contour plots. The blue dashed line represents the travel path of a vehicle at each velocity field. The difference between the departure time and the arrival time is the travel time, which is marked as a blue line on the upper horizontal axis.

minute, for both freeways and in both peak and off-peak periods. For longer horizons, the performance of the proposed method is comparable to that of ANN and SVR, whereas it always performs better than k-NN and *inst.* in these results.

First of all, it is surprising that the proposed method has comparable errors with that of ANN and SVR for longer horizons. The ANN and SVR are direct methods, which means that they have a separate model for each horizon and each one has been trained independently. The proposed method, on the other hand, is an indirect method, which predicts the travel time of longer horizons based on previous predictions. In other words, it uses a model trained only once for all the horizons.

One could understand this from its superior performance at the 0-minute horizon. As it is seen in all the sub-figures of Fig. 5, the proposed method starts from a very small error, and then the error starts to increase gradually when extending the prediction horizon. This is simply due to the aggregate noise in Eq. (14). From the fact that covariance of the sum of two Gaussian random variables is always greater than the

variance of each variable, the sum of the noise terms in Eq. (16) always produces larger covariance and therefore more substantial errors. However, since its initial 0-minute horizon error is very small compared to the other methods, the errors can remain relatively small even when the noise propagates and accumulates with time.

Compared to the other indirect methods, which are k-NN and instantaneous travel time forecaster, the proposed method shows better prediction regardless of traffic profiles and prediction horizons. We can find the reason by looking into the type of data that are considered in each method. The k-NN is highly dependent on historical data, whereas the instantaneous travel time forecaster uses only the real-time traffic measurement. Our prediction algorithm (Eq. (20)), on the other hand, utilizes both the real-time measurement ($\mathbf{v}_k$) and the historical information that is considered in the transition matrix ($\bar{H}_k$). This explains why it outperforms the other two methods.

Table II shows the mean average percentage errors (MAPE) in the peak periods and the corresponding *improvement rates*,
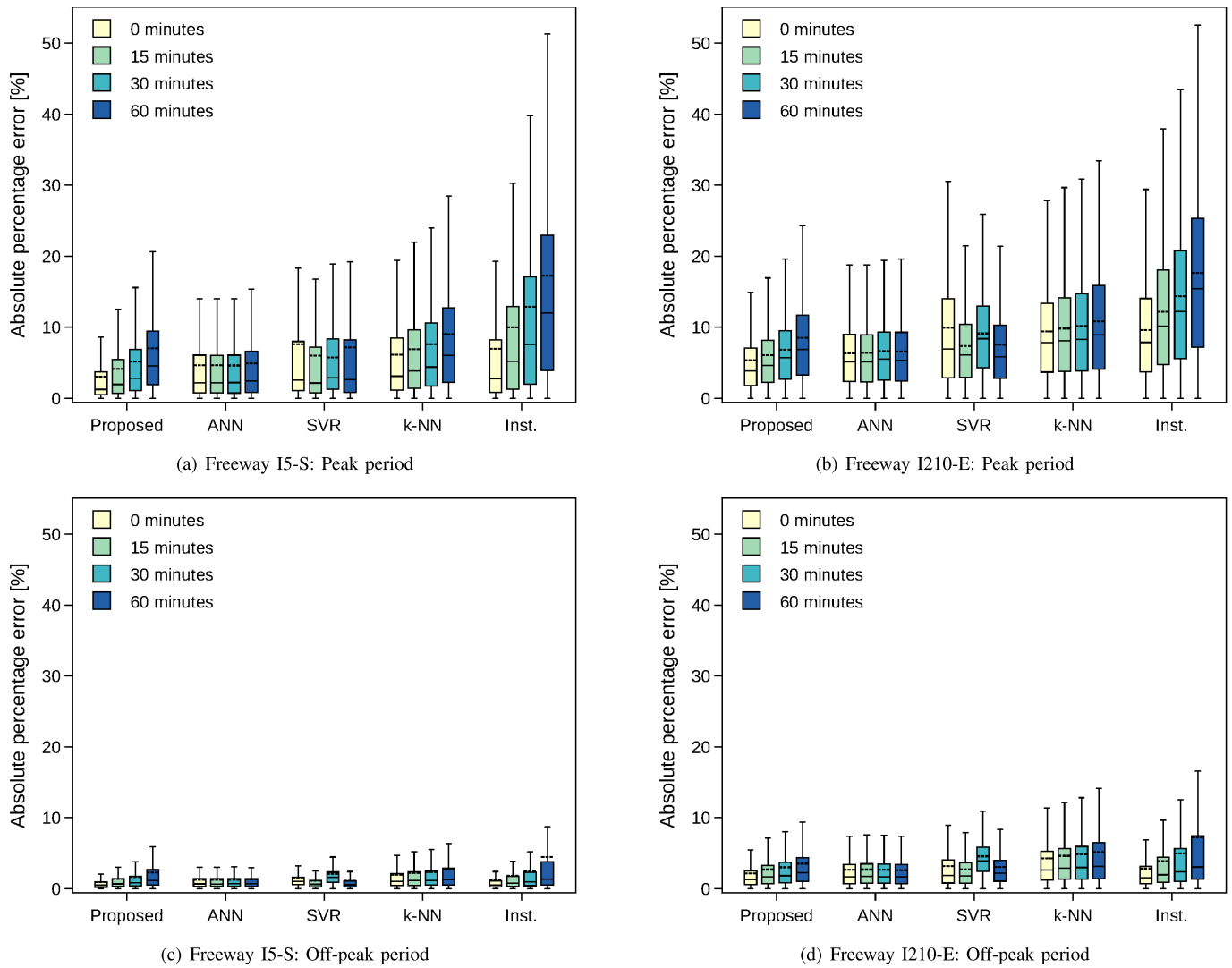
Fig. 5.   Absolute percentage errors (APE) of 5 different travel time forecasters with various horizons. Two examples of freeways in California, I5-S and I210-E, are studied during their peak periods (a) and (b); and their off-peak periods (c) and (d). Inside the box plots, the medians and mean values are marked as solid and dashed bars, respectively; different colors represent different prediction horizons.

TABLE II

MEAN ABSOLUTE PERCENTAGE ERROR ON TEST SETS FOR EACH FREEWAY

(a) Freeway I5-S

| Prediction horizon | 0 minutes | | 15 minutes | | 30 minutes | | 60 minutes | |
	MAPE	Improvement rate	MAPE	Improvement rate	MAPE	Improvement rate	MAPE	Improvement rate
Proposed	3.07	**0.56**	4.15	**0.58**	5.15	0.60	7.03	0.59
ANN	4.68	0.33	4.64	0.54	4.62	**0.64**	4.92	**0.71**
SVR	7.59	-0.09	6.02	0.40	5.77	0.55	7.18	0.58
k-NN	6.14	0.12	6.89	0.31	7.63	0.41	9.00	0.48
Inst.	6.97	-	9.99	-	12.86	-	17.26	-

(b) Freeway I210-E

| Prediction horizon | 0 minutes | | 15 minutes | | 30 minutes | | 60 minutes | |
	MAPE	Improvement rate	MAPE	Improvement rate	MAPE	Improvement rate	MAPE	Improvement rate
Proposed	5.08	**0.47**	5.76	**0.54**	6.49	0.57	8.09	0.60
ANN	6.21	0.35	6.25	0.50	6.48	**0.57**	6.37	**0.68**
SVR	9.62	-0.01	6.96	0.44	8.57	0.44	7.18	0.64
k-NN	9.07	0.05	9.49	0.24	9.77	0.36	10.29	0.49
Inst.	9.55	-	12.46	-	15.24	-	20.17	-

which indicate how much the accuracy of travel time prediction is improved compared to that of the instantaneous travel time. For instance, in the case of Freeway I5-S during the peak periods, according to Table II (a), the proposed method

improves the prediction accuracy compared to instantaneous travel time by 56% with the 0-minutes horizon. In contrast, ANN and SVR improve that by 33% and -9%, respectively.

Table II also confirms that the proposed method has the best prediction accuracy among all five forecasters for short horizons ($h = 0, 15$ min) and comparable performances to the best one for longer horizons ($h = 30, 60$ min). This result is promising since our approach has an additional degree of freedom to be used for arbitrary departure time and various starting points.

IV. CONCLUSIONS

In this work, we propose a dynamic linear model with time-varying coefficients to predict travel time. The time-varying coefficients allow the linear model to represent non-linear traffic behaviors. The transition matrix consisting of these coefficients is estimated as a least-squares solution, which can be solved analytically and thus computationally efficient. The travel time predictor based on the proposed model outperforms other predictors for short-term prediction regardless of traffic situations. This can be useful to many applications, such as car navigation systems and traffic management.

APPENDIX A
MATHEMATICAL DERIVATIONS

A. Regularized Least Squares Solution

A derivative of a scalar function is:

$$df(x, y) = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy. \tag{A.26}$$

This can be extended to a matrix form as follows:

$$df(H) = df\left(H_{1,1}, H_{2,1}, \cdots H_{n,m}\right) \tag{A.27}$$

$$= \frac{\partial f}{\partial H_{1,1}} dH_{1,1} + \frac{\partial f}{\partial H_{2,1}} dH_{2,1} + \cdots \frac{\partial f}{\partial H_{n,m}} dH_{n,m} \tag{A.28}$$

$$= vec^{\top}(dH) \cdot vec\left(\frac{df}{dH}\right) \tag{A.29}$$

$$= \mathrm{tr}\left(dH^{\top} \frac{df}{dH}\right), \tag{A.30}$$

where the function $vec(\cdot)$ vectorizes a matrix by concatenating its columns and

$$\frac{df}{dH} = \begin{bmatrix} \frac{\partial f}{\partial H_{1,1}} & \frac{\partial f}{\partial H_{1,2}} & \cdots & \frac{\partial f}{\partial H_{1,m}} \\ \frac{\partial f}{\partial H_{2,1}} & \frac{\partial f}{\partial H_{2,2}} & \cdots & \frac{\partial f}{\partial H_{2,m}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial H_{n,1}} & \frac{\partial f}{\partial H_{n,2}} & \cdots & \frac{\partial f}{\partial H_{n,m}} \end{bmatrix}, \tag{A.31}$$

$$dH = \begin{bmatrix} dH_{1,1} & dH_{1,2} & \cdots & dH_{1,m} \\ dH_{2,1} & dH_{2,2} & \cdots & dH_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ dH_{n,1} & dH_{n,2} & \cdots & dH_{n,m} \end{bmatrix}, \tag{A.32}$$

where $H_{i,j}$ denotes the i, j entry of matrix H.

We define the cost function of Eq. (8) as f:

$$f(H_k) \triangleq \rho \lambda^{|\mathbb{D}|} \|H_k\|_F^2 + \left\| \left(V_{k+1}^{\mathbb{D}} - H_k V_k^{\mathbb{D}} \right) \Lambda_{|\mathbb{D}|}^{\frac{1}{2}} \right\|_F^2. \tag{A.33}$$

Since the cost function is convex [39], we utilize that the derivative at global minimum is zero. Therefore, we compute:

$$f(H_k + dH_k) - f(H_k)$$
$$= \mathrm{tr}\left(2\left(H_k V_k^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} - V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} \right) dH_k^{\top} \right.$$
$$\left. + H.O.T. \right) + \rho \lambda^{|\mathbb{D}|} \mathrm{tr}\left(2 H_k dH_k^{\top} + H.O.T. \right). \tag{A.34}$$

Here, the abbreviation *H.O.T.* stands for higher order terms of dH_k. Assuming that dH_k is small enough,

$$f(H_k + dH_k) - f(H_k)$$
$$= df(H_k)$$
$$= \mathrm{tr}\left(2\left(H_k V_k^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \rho \lambda^{|\mathbb{D}|} H_k \right.\right.$$
$$\left.\left. - V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} \right) dH_k^{\top} \right). \tag{A.35}$$

The higher order terms are ignored since they are much smaller than the first order term dH_k. By Eq. (A.30), it is confirmed that:

$$\frac{df}{dH_k} = 2\left(H_k V_k^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} \right.$$
$$\left. + \rho \lambda^{|\mathbb{D}|} H_k - V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} \right). \tag{A.36}$$

We set the derivative equal to zero to find the global minimum, then finally:

$$\bar{H}_{k+1,k}^{\mathbb{D}} = V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} \left(V_t^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \rho \lambda^{|\mathbb{D}|} I_M \right)^{-1}. \tag{A.37}$$

B. Regularization Parameter

The data matrix $V_t^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top}$ in Eq. (10) can be decomposed as:

$$V_t^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} = U D U^{\top}, \tag{A.38}$$

where $UU^{\top} = U^{\top}U = I_M$ and D is a diagonal matrix since the matrix is symmetric. Then, we can rewrite the inner part of the inversion in Eq. (10) as follows:

$$V_t^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \rho \lambda^{|\mathbb{D}|} I_M = U \left(D + \rho \lambda^{|\mathbb{D}|} I_M \right) U^{\top}. \tag{A.39}$$

Equation (A.39) proves that even if the number of training data is not enough (i.e., there are some zero values in the diagonal of D), the inversion is still available since the regularization term $\left(\rho \lambda^{|\mathbb{D}|} I_M \right)$ is added and makes the regularized diagonal matrix $\left(D + \rho \lambda^{|\mathbb{D}|} I_M \right)$ all non-zero on the diagonal (= full rank). Therefore, the regularization term allows the model to be reliable in the inversion process.

C. Recursive Update

We define two matrices $G_k^{\mathbb{D}}$ and $P_k^{\mathbb{D}}$ as follows:

$$G_k^{\mathbb{D}} \triangleq V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top}, \tag{A.40}$$

$$P_k^{\mathbb{D}} \triangleq \left(V_k^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \rho \lambda^{|\mathbb{D}|} I_M \right)^{-1}. \tag{A.41}$$

Then we rewrite Eq. (10) with the multiplication of these two matrices:

$$\bar{H}_k^{\mathbb{D}} = G_k^{\mathbb{D}} P_k^{\mathbb{D}}. \tag{A.42}$$

The matrix $G_k^{\mathbb{D} \cup \tilde{d}}$ with a new day $\tilde{d}$, which does not belong to the training set $\mathbb{D}$, can be written as:

$$G_k^{\mathbb{D} \cup \tilde{d}} = V_{k+1}^{\mathbb{D} \cup \tilde{d}} \Lambda_{\left| \mathbb{D} \cup \tilde{d} \right|} \left(V_k^{\mathbb{D} \cup \tilde{d}} \right)^{\top} \tag{A.43}$$

$$= \left[V_{k+1}^{\mathbb{D}} \quad \mathbf{v}_{k+1}^{\tilde{d}} \right] \begin{bmatrix} \lambda \Lambda_{|\mathbb{D}|} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left(V_k^{\mathbb{D}} \right)^{\top} \\ \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \end{bmatrix} \tag{A.44}$$

$$= \lambda V_{k+1}^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \mathbf{v}_{k+1}^{\tilde{d}} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \tag{A.45}$$

$$= \lambda G_k^{\mathbb{D}} + \mathbf{v}_{k+1}^{\tilde{d}} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \tag{A.46}$$

and

$$P_k^{\mathbb{D} \cup \tilde{d}} = \left(V_k^{\mathbb{D} \cup \tilde{d}} \Lambda_{\left| \mathbb{D} \cup \tilde{d} \right|} \left(V_k^{\mathbb{D} \cup \tilde{d}} \right)^{\top} + \rho \lambda^{\left| \mathbb{D} \cup \tilde{d} \right|} I_M \right)^{-1} \tag{A.47}$$

$$= \left(\left[V_k^{\mathbb{D}} \ \mathbf{v}_k^{\tilde{d}} \right] \begin{bmatrix} \lambda \Lambda_{|\mathbb{D}|} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \left(V_k^{\mathbb{D}} \right)^{\top} \\ \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \end{bmatrix} + \rho \lambda^{\left| \mathbb{D} \cup \tilde{d} \right|} I_M \right)^{-1} \tag{A.48}$$

$$= \left(\lambda \left(V_k^{\mathbb{D}} \Lambda_{|\mathbb{D}|} \left(V_k^{\mathbb{D}} \right)^{\top} + \rho \lambda^{|\mathbb{D}|} I_M \right) + \mathbf{v}_k^{\tilde{d}} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \right)^{-1} \tag{A.49}$$

$$= \left(\lambda \left(P_k^{\mathbb{D}} \right)^{-1} + \mathbf{v}_k^{\tilde{d}} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} \right)^{-1} \tag{A.50}$$

$$= \lambda^{-1} P_k^{\mathbb{D}} - \frac{\lambda^{-1} P_k^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} P_k^{\mathbb{D}} \lambda^{-1}}{1 + \lambda^{-1} \left(\mathbf{v}_k^{\tilde{d}} \right)^{\top} P_k^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}}}, \tag{A.51}$$

where the derivations (A.50) to (A.51) are based on the matrix inversion lemma [40]. Eq. (A.46) and (A.51) show the availability of updating the matrices $\left\{ P_k^{\mathbb{D}}, G_k^{\mathbb{D}} \right\}$ to $\left\{ P_k^{\mathbb{D} \cup \tilde{d}}, G_k^{\mathbb{D} \cup \tilde{d}} \right\}$ with new measurements $\left\{ \mathbf{v}_k^{\tilde{d}}, \mathbf{v}_{k+1}^{\tilde{d}} \right\}$. Therefore, we can estimate the new transition matrices $\bar{H}_k^{\mathbb{D} \cup \tilde{d}}$ with the updated matrices $\left\{ P_k^{\mathbb{D} \cup \tilde{d}}, G_k^{\mathbb{D} \cup \tilde{d}} \right\}$ as in Eq. (A.42).

D. Linear Minimum Mean Square Estimator

A linear estimator for the velocity vector of i-step ahead at time t is written as:

$$\mathbf{v}_{k+i|k}^{\tilde{d}} = A_0 \mathbf{v}_k^{\tilde{d}} + A_1 \mathbf{v}_{k-1}^{\tilde{d}} + \cdots + A_{k-1} \mathbf{v}_1^{\tilde{d}} \tag{A.52}$$

What we need to do is to find an optimal set $\{A_0, A_1, \cdots, A_{k-1}\}$ in the minimum mean square

error (MMSE) sense. From Eq. (A.52) and Eq. (14), we define the prediction error as follows:

$$\mathbf{v}_{k+i}^{\tilde{d}} - \mathbf{v}_{k+i|k}^{\tilde{d}} = \left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} - A_0 \right) \mathbf{v}_k^{\tilde{d}} \\ - \sum_{j=1}^{k-1} A_j \mathbf{v}_{k-j}^{\tilde{d}} + \mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} \tag{A.53}$$

Its mean square is

$$\mathrm{E} \left[\left(\mathbf{v}_{k+i}^{\tilde{d}} - \mathbf{v}_{k+i|k}^{\tilde{d}} \right) \left(\mathbf{v}_{k+i}^{\tilde{d}} - \mathbf{v}_{k+i|k}^{\tilde{d}} \right)^{\top} \right]$$

$$= \mathrm{E} \left[\left[\left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} - A_0 \right) \mathbf{v}_k^{\tilde{d}} - \sum_{j=1}^{k-1} A_j \mathbf{v}_{k-j}^{\tilde{d}} + \mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} \right] \right. $$
$$\left. \left[\left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} - A_0 \right) \mathbf{v}_k^{\tilde{d}} - \sum_{j=1}^{k-1} A_j \mathbf{v}_{k-j}^{\tilde{d}} + \mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} \right]^{\top} \right]$$

$$= \mathrm{E} \left[\left[\left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} - A_0 \right) \mathbf{v}_k^{\tilde{d}} - \sum_{j=1}^{k-1} A_j \mathbf{v}_{k-j}^{\tilde{d}} \right] \right.$$
$$\left. \left[\left(\bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} - A_0 \right) \mathbf{v}_k^{\tilde{d}} - \sum_{j=1}^{k-1} A_j \mathbf{v}_{k-j}^{\tilde{d}} \right]^{\top} \right]$$

$$+ \mathrm{E} \left[\mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} \left(\mathbf{n}_{k+i-1 \leftarrow k}^{\tilde{d}} \right)^{\top} \right] \tag{A.54}$$

and Eq. (A.54) is minimized when

$$A_0 = \bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} \tag{A.55}$$

and

$$A_j = 0 \quad \text{for } j = 1, 2, \cdots, k-1. \tag{A.56}$$

Therefore, the linear MMSE estimator for the velocity vector of i-step ahead at time step k is:

$$\mathbf{v}_{k+i|k}^{\tilde{d}} = A_0 \mathbf{v}_k^{\tilde{d}} \tag{A.57}$$

$$= \bar{H}_{k+i-1 \leftarrow k}^{\mathbb{D}} \mathbf{v}_k^{\tilde{d}}, \tag{A.58}$$

which is equivalent to Eq. (18).

APPENDIX B
DIFFERENT PREDICTORS

In this section, we explain different travel time predictors which are compared with the proposed method.

A. Instantaneous Travel Time

Instantaneous travel time is calculated based on the assumption that the current state does not change with time, i.e.,

$$v \left(t', x \right) = v \left(t, x \right), \quad \forall t' > t, \ \forall x, \tag{B.59}$$

when the current time is t. The travel time $itt(t)$ is then estimated based on this velocity field with Algorithm 1.

B. k-Nearest Neighbor

The k-Nearest neighbor (k-NN) method estimate unknown velocity field with the k most similar (or nearest) days in the training set up to a current time t in terms of euclidean distance. Specifically, the velocity field for the rest of the day (after the current time) is estimated as the average of the velocities of the k nearest neighbors.. We set $k = 1$, meaning that we choose the most similar day in the training set for prediction. The travel time based on the nearest neighbor method is calculated by Algorithm 1.

C. Support Vector Regression

To implement a support vector regression (SVR) method, we have followed the same procedure of the previous work [19]. However, instead of using actual travel times as an input, we put instantaneous travel times because we think that we don't know the actual travel time at the time of estimation. We used the past 5 instantaneous travel times, i.e., $itt(t-4)$, $itt(t-3)$, …, $itt(t)$ as input variables. These input variables are scaled to have a zero mean and a unit variance. We set the target as the actual travel time with prediction horizon h, $a(t+h)$. Like in Ref. [19], the linear kernel was chosen with the parameter setting $C = 1000$ and $\tau = 0.1$.

D. Artificial Neural Network

We have designed a simple vanilla artificial neural network (ANN) for comparison using the same scaled input and target variables as in the SVR above. We set one hidden layer with 10 neurons. We used the MLPRegressor module of Scikit-learn python package and set all the parameter settings as the default setting except for the aforementioned hidden layer setting.

REFERENCES

[1] M. Bai, Y. Lin, M. Ma, and P. Wang, "Travel-time prediction methods: A review," in *Smart Computer Communication*, M. Qiu, Ed. Cham, Switzerland: Springer, 2018, pp. 67–77.

[2] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, "Network state estimation and prediction for real-time traffic management," *Netw. Spatial Econ.*, vol. 1, nos. 3–4, pp. 293–318, Sep. 2001.

[3] N. Wan, G. Gomes, A. Vahidi, and R. Horowitz, "Prediction on travel-time distribution for freeways using online expectation maximization algorithm," in *Proc. 93rd Annu. Meeting*, 2014, p. 3221.

[4] S. Takaba, T. Morita, T. Hada, T. Usami, and M. Yamaguchi, "Estimation and measurement of travel time by vehicle detectors and license plate readers," in *Proc. Conf. Vehicle Navigat. Inf. Syst.*, vol. 2, Oct. 1991, pp. 257–267.

[5] A. Skabardonis and N. Geroliminis, "Real-time estimation of travel times along signalized arterials," in *Proc. Transp. Traffic Theory (ISTTT)*. Riverport Lane, MD, USA: Elsevier, 2005, pp. 387–406.

[6] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1855, no. 1, pp. 49–59, Jan. 2003.

[7] J. Rice and E. Van Zwet, "A simple and effective method for predicting travel times on freeways," in *Proc. Intell. Transp. Syst.*, Oct. 2001, pp. 227–232.

[8] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 11, nos. 3–4, pp. 187–210, Jun. 2003.

[9] M. Yang, Y. Liu, and Z. You, "The reliability of travel time forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 162–171, Mar. 2010.

[10] J. Xia, M. Chen, and W. Huang, "A multistep corridor travel-time prediction method using presence-type vehicle detector data," *J. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 104–113, May 2011.

[11] Y. Zhang, A. Haghani, and X. Zeng, "Component GARCH models to account for seasonal patterns and uncertainties in travel-time prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 719–729, Feb. 2015.

[12] A. Salamanis, D. D. Kehagias, C. K. Filelis-Papadopoulos, D. Tzovaras, and G. A. Gravvanis, "Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1678–1687, Jun. 2016.

[13] S. I.-J. Chien and C. M. Kuchipudi, "Dynamic travel time prediction with real-time and historic data," *J. Transp. Eng.*, vol. 129, no. 6, pp. 608–616, Nov. 2003.

[14] C. M. Kuchipudi and S. I. J. Chien, "Development of a hybrid model for dynamic travel-time prediction," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1855, no. 1, pp. 22–31, Jan. 2003.

[15] J. W. C. Van Lint, "Online learning solutions for freeway travel time prediction," *IEEE Transp. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 38–47, Jan. 2008.

[16] A. Achar, D. Bharathi, B. A. Kumar, and L. Vanajakshi, "Bus arrival time prediction: A spatial Kalman filter approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1298–1307, Mar. 2020.

[17] C. P. I. van Hinsbergen, J. W. C. van Lint, and H. J. van Zuylen, "Bayesian committee of neural networks to predict travel times with confidence intervals," *Transp. Res. C, Emerg. Technol.*, vol. 17, no. 5, pp. 498–509, Oct. 2009.

[18] X. Fei, C.-C. Lu, and K. Liu, "A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 19, no. 6, pp. 1306–1318, Dec. 2011.

[19] C.-H. Wu, C.-C. Wei, D.-C. Su, M.-H. Chang, and J.-M. Ho, "Travel time prediction with support vector regression," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2003, pp. 1438–1442.

[20] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[21] P. Gao, J. Hu, H. Zhou, and Y. Zhang, "Travel time prediction with immune genetic algorithm and support vector regression," in *Proc. 12th World Congr. Intell. Control Autom. (WCICA)*, Jun. 2016, pp. 987–992.

[22] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 14, no. 5, pp. 357–367, Sep. 1999.

[23] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *Eur. J. Oper. Res.*, vol. 131, no. 2, pp. 253–261, Jun. 2001.

[24] A. Dharia and H. Adeli, "Neural network model for rapid forecasting of freeway link travel time," *Eng. Appl. Artif. Intell.*, vol. 16, nos. 7–8, pp. 607–613, Oct. 2003.

[25] S. Innamaa, "Short-term prediction of travel time using neural networks on an interurban highway," *Transportation*, vol. 32, no. 6, pp. 649–669, Nov. 2005.

[26] J. W. C. van Lint, S. P. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. C, Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, Oct. 2005.

[27] H. Liu, H. Van Zuylen, H. Van Lint, and M. Salomons, "Predicting urban arterial travel time with state-space neural networks and Kalman filters," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1968, no. 1, pp. 99–108, Jan. 2006.

[28] C.-S. Li and M.-C. Chen, "A data mining based approach for travel time prediction in freeway with non-recurrent congestion," *Neurocomputing*, vol. 133, pp. 74–83, Jun. 2014.

[29] Y. Hou and P. Edara, "Network scale travel time prediction using deep learning," *Transp. Res. Rec.*, vol. 2672, no. 45, pp. 115–123, 2018.

[30] Y. Duan, Y. L. V., and F.-Y. Wang, "Travel time prediction with LSTM neural network," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1053–1058.

[31] Y. Liu, Y. Wang, X. Yang, and L. Zhang, "Short-term travel time prediction by deep learning: A comparison of different lstm-dnn models," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.

[32] B. Hamner, "Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow," in *Proc. IEEE Int. Conf. Data Mining Workshops*, Dec. 2010, pp. 1357–1359.

[33] Y. Zhang and A. Haghani, "A gradient boosting method to improve travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 308–324, Sep. 2015.

[34] N. C. Petersen, F. Rodrigues, and F. C. Pereira, "Multi-output bus travel time prediction with convolutional LSTM neural network," *Expert Syst. Appl.*, vol. 120, pp. 426–435, Apr. 2019.

[35] S. Robinson and J. W. Polak, "Modeling urban link travel time with inductive loop detector data by using the k-NN method," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1935, no. 1, pp. 47–56, Jan. 2005.

[36] M. Yildirimoglu and N. Geroliminis, "Experienced travel time prediction for congested freeways," *Transp. Res. B, Methodol.*, vol. 53, pp. 45–63, Jul. 2013.

[37] T. Kailath, B. Hassidi, and A. H. Sayed, *Linear Estimation*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[38] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 2. Hoboken, NJ, USA: Wiley, 2008.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[40] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3. Baltimore, MD, USA: JHU Press, 2012.

Semin Kwak (Member, IEEE) received the B.Sc. degree in electrical engineering from Yonsei University, Seoul, South Korea, in 2013, and the M.Sc. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering with the Urban Transport Systems Laboratory (LUTS), École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. His current research interests include time-series forecasting, graph theory, machine learning, and optimization with applications of transportation systems.

Nikolas Geroliminis (Member, IEEE) received the Diploma degree in civil engineering from the National Technical University of Athens, and the M.Sc. and Ph.D. degrees in civil engineering from the University of California at Berkeley, Berkeley, CA, USA. He is currently an Associate Professor with the École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, and the Head of the Urban Transport Systems Laboratory. His research interests include urban transportation systems, traffic flow theory and control, public transportation and on-demand transportation, optimization, and large scale networks. He is a member of the Transportation Research Board's Traffic Flow Theory Committee. He also serves as an Associate Editor of *Transportation Research Part C*, *Transportation Science*, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and in the Editorial Board of *Transportation Research Part B* and *Transportmetrica B*.