A Travel Time Prediciton with Machine Learning Algorithms

Younggeun Cho and Jungsuk Kwac
Department of Electrical Engineering
Stanford University
Stanford, CA, 94305, USA
E-mail: ygcho@stanford.edu,kwjusu1@stanford.edu;

I. Introduction

It is Friday afternoon. You have a blind date at San Francisco at 8:00pm. Since you don't want to give a bad first impression, you plan to leave early enough not to be late for the appointment. However, the traffic on 101North on every Friday is real heavy and unpredictable. Although it takes only 50 minutes to get there on other weekdays, it might take from 1.5 hours to 3 hours. So, when should you leave for SF to minimize the wasted time and meet the appointment schedule?

This question above is commonly encountered by travelers, so they want an accurate travel time prediction. Adding to this case, accurate travel time prediction is also essential for traffic information and transportation systems such as vehicular navigation systems in cars or traffic boards on the roads to provide efficient route information. Furthermore, with this information, we can prevent unnecessary traffic jams with such a scheme as load balancing. In this project, we try to predict a travel time from one place to another in a freeway using machine learning algorithms.

There are several research papers dealing with this topic. Especially, J. M. Kwon et al [3] and X. Zhang et al [2] form a good starting point. These previous works provide several travel time prediction algorithms using linear regression. For these algorithms, inputs for linear regression are historical data of traveling times and current states of highways. However, the authors for these papers did not consider other important factors such as what day of the week and which season of the year. By considering additional important factors, we expect to improve the accuracy of prediction. The purposes of this project are twofold; first, we evaluate existing algorithms presented in relevant research papers; second, we develop new algorithms to enhance the accuracy of prediction.

For evaluation and development of travel time prediction algorithms, we need several types of data including an average speed of each freeway for different times, days, and months. These data are available to public at Freeway Performance Measurement System website [1]. This website provides historical and real-time freeway data from freeways in the State of California collected by sensors installed on major highways.

II. Problem Statements

As in [2],the travel time prediction problem is defined as follows: The route for traveling comprises L segments of freeway with length $S_1, ..., S_L$. In each segment, there is a sensor for measuring the average speed of bypassing vehicles, $v(l, \tau_i)$, where τ_i is the time for measurement. It is assumed that for the prediction at time t, the speed data, $v(l, \tau_i)$, are available for $\tau_i < t - \Delta$, where Δ is a fixed positive time-lag. The objective of this project is to predict the travel time of this route with this information. More formally, we should find a function g such that

$$\hat{T}(t) = g(V(t, \Delta)), \tag{1}$$

where $\hat{T}(t)$ is a time predict for travel that starts at time t and $V(t, \Delta) = [v(i, \tau_i), l = 1, ..., L, \tau_i \leq t - \Delta]$ denotes the collection of data available for predicting $T(t)$, the real travel time. The performance metric for prediction is the percentage prediction error defined as

$$\gamma(t) = \frac{|T(t) - \hat(T)(t)|}{T(t)}. \tag{2}$$

III. Prediction Methods

Travel time algorithms differ in choosing the predictor function $g(V(t, \Delta))$ in (1). In this project, several prediction methods are evaluated including the time-varying coefficient linear model of [2].

A. Historical means(HM)

Historical mean (HM) uses the mean of the training set at time t as the predictor for travel time. Formally,

$$\hat{T}_{HM}(t) = \frac{1}{|D|} \sum_{d \in D} T_d(t).$$

B. Current time predictor(CT)

For *current time predictor(CT)* method, the travel predictor is obtained by assuming the speeds of all segments at time t remain the same as the speeds at $t - \Delta$ throughout the entire travel. Formally,

$$\hat{T}_{CT}(t) = \sum_{l=1}^{L} \frac{S_l}{v(l, t - \Delta)}. \tag{3}$$

C. Time varying linear regression(TVLR)

In this method, the travel time predictor is expressed as

$$\hat{T}_{TVLR} = \alpha(t,\Delta) + \beta(t,\Delta)\hat{T}_{CT}(t), \tag{4}$$

where $alpha(t,\Delta)$ and $beta(t,\Delta)$ are time-varying constant, and $T_{CT}(t)$ is the current travel time predictor of the previous segment.

D. Time varying linear regression per segment(TVPS)

In TVLR, the predictor is obtained by considering the current time predictor of the entire route as the key feature. For *time varying linear regression per segment(TVPS)*, the current time predictor of each segment plays a role as a feature. So, linear regression coefficients, $alpha$ and $beta$, are calculated for each segment separately so that travel time prediction is conducted on each segment. Then, predictors for all segments are summed up to form the whole predictor as

$$\hat{T}_{TVPS}(t) = \sum_{l=1}^{L} \hat{T}_{TVLR}(t,l) \sum_{l=1}^{L} \alpha(t,l) + \beta(t,l)\hat{T}_{CT}(t,l).$$

E. Algorithm considering Days in a Week

In the previous papers [2] and [3], the authors obtained $\alpha(t,\Delta)$ and $\beta(t,\Delta)$, and predicted $\hat{T}(t)$ assuming the traveling time characteristics of Fridays are the same as those of Mondays, for example. However, as shown in Fig. 1, the statistical characteristics for each day in a week differ significantly.

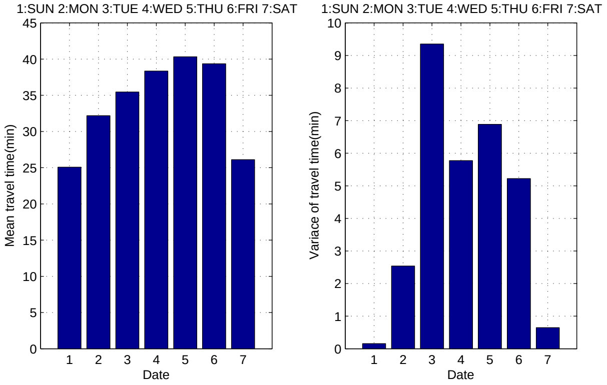

Fig. 1. Means and variances of travel time for a trip from University Avenue to San Francisco with Highway 101N.

The new algorithm exploits this difference in such a way that the coefficient for each day is obtained by training on a subset of S that contains only the same days in a week. More formally, $\alpha_D(t,\Delta)$ and $\beta_D t,\Delta$ minimize

$$\begin{aligned}&\epsilon_{S_D}(\alpha_D(t,\Delta),\beta_D(t,\Delta))\\ &= \sum_{s_n \in S} \big(T(t,s_n) - \alpha(t,\Delta) - \beta(t,\Delta)\big)^2,\end{aligned}$$

where $D \in \{SUN, MON, TUE, WED, THU, FRI, SAT\}$ and $S_D = \{s_i \in S; s_n \text{ collected on } D \text{ day in a week}\}$. Once obtained, $\alpha_D(t,\Delta)$ and $\beta_D t,\Delta$ is used to predict the travel times only for D days.

IV. Data Acquisition and Preprocessing

A. Data Acquisition

We get the data from the Freeway Performance Measurement System website[1]. For this milestone report, trip starts at University ave to San Francisco on highway 101N. There are 68 VDSs(Vehicle Detection System) in the route. The speed data measured by VDSs for every 5 minutes. That website supports exporting data as excel or text file, and we used 'unix' command in Matlab and 'wget' command in Unix to get data.

B. Preprocessing

Before training our model, preprocessing the travel time from the data is necessary. To calculate the whole travel time taken from University ave to San Francisco, the travel time is summed over all small segments between VDSs. However, two problems should be addressed before getting the travel time. First, the entire route should be re-segmented to reflect the real speed of the vehicles. Second, the fact that the time is also discretized should be considered.

For the first problem, re-segmentation, consider the segmented route in Fig. 2. $x_1, ..., x_L$ are the locations of sensors(e.g. VDS in PeMS) that measure the vehicle speed. The issue here is to decide how $v(l,t)$, the speed measured at x_l, can be used. It is reasonable to assume that the speed measure at x_l represents the average speed of segment from $(x_{l-1} + x_l)/2$ to $(x_l + x_{l+1})/2$ as shown in Fig. 2. So, the segment lengths $S_l, l = 1, ..., L$ associated with speeds $v(t,l), l = 1, ..., L$, should be calculated as

$$\begin{aligned}&S_1 = (x_1 + x_2)/2 - x_1,\\ &S_l = \frac{x_l + x_{l+1}}{2} - \frac{x_{l-1} + x_l}{2}, \text{ for } l = 2, ..., l-1,\\ &S_L = x_L - (x_{L-1} + x_L)/2 \qquad .\end{aligned}$$

Then, by assuming that $v(t,l)$ remains the same when the vehicle is in the segment l, we can calculate the travel time for this segment.

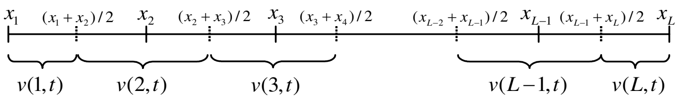

Fig. 2. Re-segmentation of the route

Once, re-segmentation is done, there remains the problem of discretized time. This problem stems from the fact that sensors measure the speeds with some time interval, δt. $\delta t = 5$ minutes for PeMS. So, if a car enters segment l at time t and stays at the same segment more than δt, it speeds should be changed to be $v(t+\delta t, l)$. Figure 3 illustrates this issue. The horizontal axis and the vertical axis are discretized so that the space-time is divided into small grids. The blue line represents the path of a vehicle. In each grid, the speed remains constant so the slope of the line is constant. The travel time can be computed the time when the path reaches the end of the last segment.

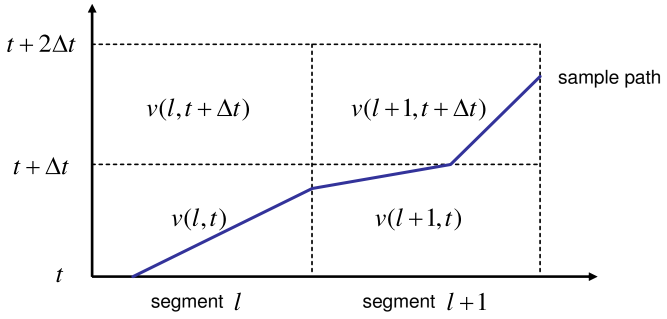

Fig. 3. Calculation of travel time for discretized time

As the authors in [2] argue, the travel times computed with this preprocessing is not significantly differ from the real travel times. So, we use the outputs of this preprocessing for training and testing of the prediction methods.

V. Numerical Results

In this section, we compare the four prediction methods in III, and apply the method considering days-in-a-week effect.

A. Test environments

As mentioned in the previous section, for numerical results the trip starts at University ave and ends at San Francisco on highway 101N. The training set consists of the trips of 209 days from April 1st to October 26th. For each day, there are 6 trips, the starting time of which is in every 5 minutes between 17:30PM to 17:55PM. We set the training and test time like this because this time slot shows the most dynamic traffics due to commuter traffic. For this trip there are 68 VDSs, which means $L = 68$. The data in PeMS shows the speeds for each lane in the highways. We assume that the speeds for the second lane represent the average speeds of vehicles. The test set of the prediction methods consists of 63 days.

B. Test result

Figure 4 shows the relative error averaged over test set with $\Delta = 30$ minutes for different days in a week. As seen in the figure, we have two different results for weekdays and weekend. For weekdays, TVPS outperforms other methods, and CT follows. The performances of HM can be considered as lower bounds due to its simplicity. For weekend, CT performs better than TVPS, which is the reverse of the results for weekdays. Recall that CT perform well when the traffic patter varies slowly. Hence, for weekends, when no commuter traffic exists, CT outperforms other methods.

Although TVPS performs well in weekdays, its performance is problematic for weekend. For TVLR, the performance is worse with relative errors more than 10%. This issue can be addressed by using methods considering days-in-a-week effect. In Fig. 5 the performances of TVPS and TVLR considering days-in-a-week effect, denoted by TVPS(d) and TVLR(d), are added to Fig. 4. As shown in the figure after considering days-in-a-week effect, both TVPS(d) and TVLR(d) outperform CT. Here, the relative error of CT remains the same since days-in-a-week effect is inherently reflected in CT. As the figure shows, the average relative error is below 10% for TVPS(d) and TVLR(d). It should be noted that TVPS involves more computational complexity than TVLR by factor of L. Therefore, it is desirable to use TVLR(d) since with less computational complexity it shows comparable performance to that of TVPS(d),which has the best prediction power.

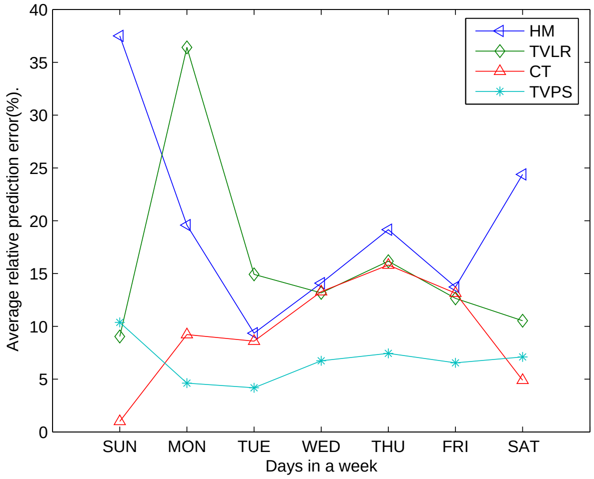

Fig. 4. Results without considering days-in-a-week effect, $\Delta = 30min$.

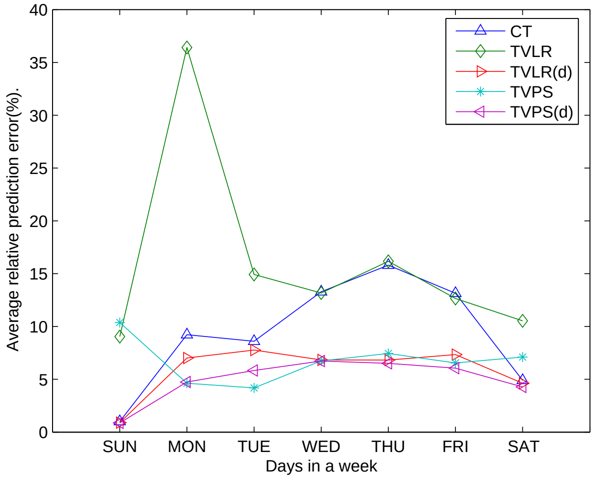

Fig. 5. Results with considering days-in-a-week effect, $\Delta = 30min$.

Figure 6 shows the effect of different values of Δ to the average relative errors of three scheme. Two things can be observed from the plot. Fist, the average relative errors of CT, TVLR(d), and TVPS(d) increase as Δ increases. Second, the performances of three methods get closer to all another as Δ increases.

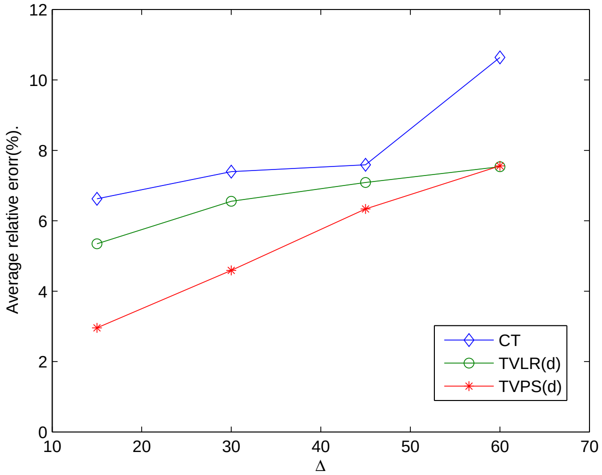

Fig. 6. Relative error vs. $Delta$.

VI. Conclusion

In this project, the problem of predicting travel time is addressed. As predictors, four methods are presented and one suggestion, 'days-in-a-week' effect, for improving the performance is made. As shown in the evaluation results with the route from Univ. Avenue to San Francisco on highway 101 North, considering 'days-in-a-week' reduces prediction error significantly. Furthermore, TVLR(d) is suggested as a useful method with its good performance and simplicity.

To further improve the accuracy of prediction, it can be considered using other models such as the linear model using link flow and occupancy. We can also try to consider other factors that affect travel time such as the traffic information for the crossing highways.

References

[1] "Freeway Performance Measurement System (PeMS)," http://pems.eecs.berkeley.edu.

[2] X. Zhang and J. Rice, "Short-Term Travel Time Prediction," *Transportation Research Part C,* Vol. 11, 2003, pp. 187-210.

[3] J. M. Kwon and K. Petty, "A Travel Time Prediction Algorithm Scalable to Freeway Networks With Many Nodes with Arbitrary Travel Routes," *Trasportation Research Record,*" 2005.

[4] J. M. Kwon and P. Varaiya, "Components of Congestion: Delay from Incidents, Special Events, Lane Closures, Weather, Potential Ramp Metering Gain, and Excess Demand", *Transportation Research Record*, 2006.