Traffic Stream Short-term State Prediction using Machine Learning Techniques

Mohammed Elhenawy, Hesham Rakha and Hao Chen
Virginia Tech Transportation Institute, 3500 Transportation Research Plaza, Blacksburg, VA 24061, U.S.A.

Keywords: Transportation Planning and Traffic Operation, Real-time Automatic Congestion Identification, Mixture of Linear Regression, ITS.

Abstract: The paper addresses the problem of stretch wide short-term prediction of traffic stream state. The problem is a multivariate problem where the responses are the speeds or flows on different road segments at different time horizons. Recognizing that short-term traffic state prediction is a multivariate problem, there is a need to maintain the spatiotemporal traffic state correlations. Two cutting-edge machine learning algorithms are used to predict the stretch-wide traffic stream traffic state up to 120 minutes in the future. Furthermore, the divide and conquer approach was used to divide the large prediction problem into a set of smaller overlapping problems. These smaller problems are solved using a medium configuration PC in a reasonable time (less than a minute), which makes the proposed technique suitable for practical applications.

1 INTRODUCTION

Nowadays, due to the technology advances, the intelligent transportation systems (ITS) are widely deployed in many countries to manage the transportation resources and solve traffic problems. Advanced traffic management systems (ATMS) and advanced traveller information systems (ATIS) are two ITS's components that are mainly involved in relaxing traffic congestion and decreasing travel time. The ATMS collect real-time traffic data using different sensing devices such as cameras and speed sensors. These collected data are fed to the Traffic Management Center (TMC) where it is fused together and get ready for downstream analysis and prediction. Based on the outcome of the analysis, actions can be taken (e.g. traffic routing, DMS messages) to avoid congestion and decrease travel time.

Data-driven modelling is considered a good approach to model complex traffic characteristics when applying mathematical models that are based on macroscopic and microscopic theories of traffic flow is difficult. Data-driven short-term prediction of the traffic characteristics such as flow, density and speed has been a very important tool in ITS. The short-term prediction is not a straightforward task because of the unstable traffic conditions and complex road settings (Vlahogianni et al., 2014).

During the last decades, the traffic characteristics prediction has been studied and many prediction approaches have been developed. The developed prediction approaches are classified into three broad categories; parametric models, nonparametric models, and simulations.

Time-series techniques is a parametric model that is used widely in traffic flow prediction. The autoregressive integrated moving average (ARIMA) model was used very early to predict short-term freeway traffic flow (Ahmed and Cook, 1979). After that, different advanced versions of ARIMA were used to develop more accurate prediction models. Voort et al. integrated the Kohonen self-organizing map and ARIMA into a new method called KARIMA(Van Der Voort et al., 1996). KARIMA uses a Kohonen self-organizing map to cluster the data and then model each cluster using ARIMA. Lee et al. used subset ARIMA model for the one-step-ahead forecasting task which gave more stable and accurate results than the full ARIMA model (Lee and Fambro, 1999).

Due to both the highly nonlinear nature of traffic characteristics and availability of data, nonparametric methods attracted the researchers' attention. In the traffic flow prediction area, there are many versions of K-NN algorithm that showed a good prediction accuracy. Davis and Nihan argue that K-NN can capture linear and nonlinear relationships therefore it

Elhenawy, M., Rakha, H. and Chen, H.
Traffic Stream Short-term State Prediction using Machine Learning Techniques.
In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2016)*, pages 124-129
ISBN: 978-989-758-185-4

is able to model the nonlinear transition between free-flow and congested traffic(Davis and Nihan, 1991).

However, the results of their empirical study showed that K-NN is not better than a simple univariate time-series forecasts. Sun et al. considered the traffic prediction model as a non-linear system which has historical and current traffic characteristics as inputs and its output is the future traffic characteristics(Sun et al., 2003). Therefore, they used the local linear regression model to approximate the nonlinear relationship between system inputs and outputs and to predict future traffic characteristics. Young-Seon et al. proposed a short-term traffic flow predictions algorithm that combines the online-based SVR with weighted learning method for short-term traffic flow predictions (Young-Seon et al., 2013). ANN is considered one of the best tools to model highly non-linear relationship between inputs and outputs so that there are many papers that adopted many ANN models for predicting traffic flow such as the Bayesian neural network (Zheng et al., 2006) and radial basis function neural network(Park et al., 1998). Interested readers are recommended to read (Vlahogianni et al., 2014) for a good review of the proposed techniques and challenges of short-term prediction.

2 PROBLEM STATEMENT

In this Paper, we are interested in the short-term prediction of stretch-wide speed/flow. The evolution of traffic state is a complex spatiotemporal process. In order to define our prediction problem, we first define the spatiotemporal state matrix V_{IxT}, where I is the number of the stretch's segments and T is the day's time intervals. The traffic state prediction problem can be stated as follows. Let v_i^t be the observed elements of the spatiotemporal traffic state matrix at the time interval $t = 1,2, \ldots, t_0$ and segment $i = 1,2, \ldots, h$ of the studied road stretch. Our goal is to predict the spatiotemporal traffic state submatrix that spans the time interval $[t_0 + 1,\ t_0 + \Delta]$ for some prediction horizon Δ, given the spatiotemporal observed traffic state submatrix that ends at time t_0, the forecasted weather condition, and the visibility level.

The problem state above has the general solution form shown in equation (1);

$$y_{t_0+\Delta} = G\left(X_{t_0}, e_{t_0}, \Theta\right) + e_{t_0+\Delta} \qquad (1)$$

Where

G	The chosen model
$y_{t_0+\Delta}$	The response at some prediction horizon Δ
X_{t_0}	The inputs predictors which includes the observed elements of the spatiotemporal traffic state submatrix at the time interval t_0
Θ	Estimated model parameters
$e_{t_0+\Delta}$	Errors (unexplained variability) because of absence of the factors that we cannot observe

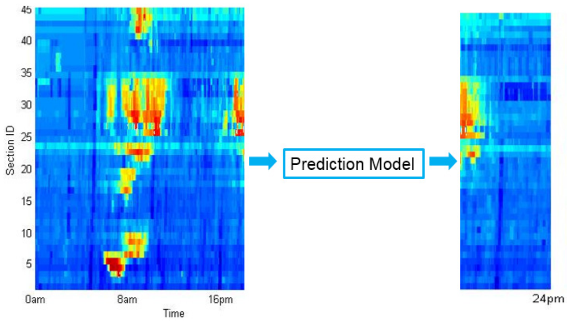

Figure 1: Prediction Model. Illustration of problem where model is needed to predict the traffic state evolution in time (x-axis) and space (y-axis).

2.1 Why Machine Learning is the Suitable Framework

Machine learning techniques are suitable models for this problem for three reasons. First is the stochastic nature of the input-output data where it is possible to find two different responses for the same input. In other words, the response corresponding to any input predictors is a distribution rather than a single point in the response space. Second, the problem is multivariate and the relationships between variables are nonlinear. Third, there is no closed mathematical form (model) that can be used to explain the relationship between the input predictors and the response.

3 METHODS

3.1 Partial Least Squares Regression (PLSR)

Multiple linear regression (MLR) is generally a good tool for modelling the relationship between predictors and responses. In many scientific problems, the relationship between the predictors and responses are poorly understood, and the main goal is to construct a good predictive model using a large number of predictors. MLR is effective when the number of predictors is small, there is no significant multicollinearity, and there is a well-understood relation between predictors and responses (Abdi). However, if the number of predictors gets too large, an MLR model will over-fit the sampled data perfectly but fail to predict new data well. Accordingly, in this case, MLR is not a suitable tool.

PLSR is a recently developed technique that generalizes and combines features from principal component analysis and MLR. It is used to predict Y from X and to describe their common structure. PLSR assumes that there are only a few latent factors that account for most of the variation in the response. The general idea of PLSR is to try to extract those latent factors, accounting for as much of the predictors' X variation as possible, and at the same time to model the responses well.

3.2 Artificial Neural Networks (ANN)

In machine learning, artificial neural networks (ANN) are used to estimate or approximate unknown linear and non-linear functions that depend on a large number of inputs. Artificial neural networks can compute values or return labels using inputs.

An ANN consists of several processing units, called neurons, which are arranged in layers. We used the multi-layered feed-forward ANN, in which the neurons are connected by directed connections, which allow information to flow directionally from the input layer to the output layer. A neuron k at layer m receives an input x_j from each neuron j at layer $m - 1$. The neuron adds the weighted sum of its inputs to a bias term. The whole thing is then applied to a transfer function and the result is passed to its output toward the downstream layer.

3.3 Principal Component Analysis (PCA)

The stretch-wide prediction problem is a multivariate problem that may involve a considerable number of correlated predictors. PCA is a popular technique for dimensionality reduction that linearly transforms possibly correlated variables into uncorrelated variables called principal components.

PCA is usually used to reduce the number of predictors involved in the downstream analysis; however, the smaller set of transformed predictors still contains most of the information (variance) in the large set. The principal components are the Eigenvectors of the dataset covariance matrix. The first principal component is the normalized Eigenvector, which is associated with the highest Eigenvalue. The first principal component represents the direction in the space that has the most variability in the data, and each succeeding component accounts for as much of the remaining variability as possible.

4 MODEL CALIBRATION

4.1 Divide and Conquer Approach

The big challenges to stretch-wide traffic state short-term prediction are the large dimension of the predictors and responses vectors and the huge number of parameters required for estimation. Once the road stretch grew to a certain point, most of the machine-learning algorithms we usually used either required too much time for training or suffered from memory problems. To handle these issues, a divide and conquer approach model was adopted in this study.

A divide and conquer paradigm suggests that if the problem cannot be solved as is, it should be decomposed it into smaller parts, and these smaller parts then solved. A divide and conquer algorithm breaks down a problem into two or more smaller problems of the same type. The final solution to the larger, more difficult problem is the combination of the smaller problems' solutions. Divide and conquer is applied in a straightforward manner to our prediction problem by dividing the inputs predictors of the spatiotemporal speed or flow matrix into smaller overlapping windows and then doing the same with the responses. The overlap of the windows is important if we need to get smooth predicted responses. Because of this overlap between windows, each segment has two predicted speeds/flows at the testing phase, and the final predicted speed/flow for overlapped segments is the average.

4.2 Training and Testing Phase

Typically, in machine learning, the model calibration process consists of a training phase and a testing phase. In the training phase, the model parameters are estimated using the training dataset. In the testing phase, the constructed models' accuracy is tested using an unseen dataset called the testing dataset.

The training phase in our approach includes the following steps:

1. Partitioning (dividing) the whole stretch into small windows, which each have a small number of segments.
2. Preparing the X and Y matrices for each window by reshaping the traffic state, weather,

and visibility inside the windows, which have widths of h and l respectively.

3. Shifting the window to the right and repeating step 2 to get another raw X and Y.
4. Applying the machine learning algorithm to the ,Y matrices to get the model parameter such as the Coefficient matrix β in the case of PLSR.

The testing phase is always simpler and does not need large time. For example, it includes multiplying the testing data by the matrix of the PLSR coefficient or passing the testing data through the neural network after reducing it using the same principal components. The last step in testing is collecting the predicted pieces together to get the prediction for the whole stretch.

5 EXPERIMENTAL ANALYSIS

5.1 Study Site

Traffic speed and flow from loop detectors are used to develop the proposed prediction models. Specifically, the study included 2013~2014 data along US-75 northbound as shown in Figure 2. This road segment includes 42 loop detectors along 23.3 miles. In order to reduce the stochastic noise and measurement error, raw speed data were aggregated by 5-minute intervals and 15-minutes interval. Therefore, the traffic speed and flow matrices over spatial (upstream to downstream) and temporal domains could be obtained for each day.

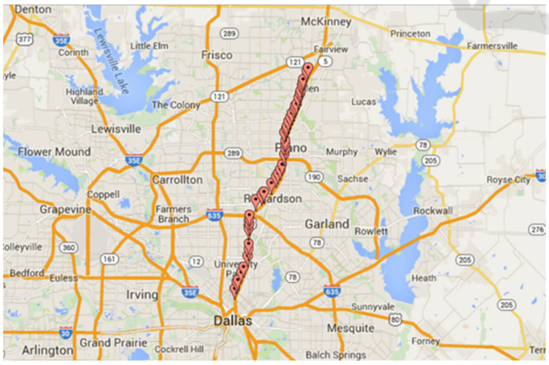

Figure 2: Layout of the Selected Freeway Stretch on US-75. (Source: Google Maps).

5.2 Evaluation Criteria

The mean absolute percentage error (MAPE) and the mean absolute error (MAE) were calculated for the different proposed algorithms

$$\mathrm{MAPE} = \frac{100}{\mathrm{IxJ}} \sum_{j=1}^{J} \sum_{i=1}^{I} \frac{\left|y_i^j - \widehat{y_i^j}\right|}{y_i^j} \tag{2}$$

$$\mathrm{MAE} = \frac{1}{\mathrm{IxJ}} \sum_{j=1}^{J} \sum_{i=1}^{I} \left|y_i^j - \widehat{y_i^j}\right| \tag{3}$$

Where
J = total number of observations in the testing data set,
I = total number of elements in each observation,
y = ground truth traffic state, and
$\hat{y}$ = predicted traffic state.

5.3 Investigating the Effect of Window Size on Prediction Errors

Our method for solving the wide stretch prediction problem is based on a divide and conquer approach, which requires fine-tuning the window size (w). In this section, we perform a sensitivity analysis of the w parameter in the divide and conquer approach. We compare the performance of the PLSR and PCA+ANN for different w values.

The ANN is a suitable technique for the stretch-wide prediction and its performance is close to PLSR's performance as will be shown; however, its training time is significantly larger compared to PLSR. In this section, to overcome the training time problem, we adopted the PCA as the dimension reduction technique. PCA is used to transform the training predictors' matrix and use the subspace consisting of the principal components with the most variance. In this paper we use the PCA to reduce the dimension of input data to 50% of its original size. Using PCA would also fix the multicollinearity problem if it existed. Moreover, we used the 15-minutes aggregated data with the PCA+ANN where using 5-minutes aggregated with this approach is very time-consuming and therefore impractical. The neural network used in this experiment has only one hidden layer which has 9 neurons. The activation function of the hidden neurons is Tanh and the activation function of the output layer is linear. In this experiment set we used the 5-minutes aggregated data when finding the best window size when using PLSR algorithm to build the prediction model. The experimental results show that as we increase the window size, the errors are reduced. Moreover, the divide and conquer approach using w=32 is the best approach, and is slightly better than the model that does not use divide and conquer.

We investigate the effect of the window size using ANN+PCA and as shown in Table 1, as we increase the window size, the errors are almost the same at small prediction horizon and are increased at

Table 1: The mean of MAPE (%) of the Us-75 flow dataset (15-minute aggregated) using ANN+PCA at different window size.

Prediction horizon (minutes)	window size 2	window size 4	window size 8	window size 16	window size 32
15	11.0840	11.3875	10.8014	10.9167	10.6331
30	12.5249	11.6274	11.8170	11.6387	11.7886
45	12.0642	12.4716	12.3469	12.6305	12.5065
60	12.6834	12.6261	12.7570	13.4120	12.7703
75	13.2808	13.0412	13.2466	13.6641	13.7153
90	13.8105	13.3066	13.5227	14.0085	14.5708
105	13.7811	13.5860	13.7143	14.3569	14.2725
120	13.9212	13.9483	14.0594	14.5721	14.8917

prediction horizon greater than or equal 90 minutes. So that we set up the window size equals to four when using ANN+PCA. In order to explain why the errors increases as the window size increase recall that large window means large neural network and large number of free parameters (coefficients and biases). Network with large number of parameters is more prone to overfitting so that network validation process stops the network training when there is no improvement in the neural network cost function during validation phase. One solution to overcome this problem is increasing the training dataset witch is not feasible in our case.

5.4 Error Reduction by Result Averaging

The results in the previous section show that the best window size for PLSR is 32; however, the MAE and MAPE are still large. For example, the MAPE for two hours of prediction is 23.34% .We visually inspected the speed and flow patterns of the five minutes aggregated data and observed two types of variations. The first variation has a low frequency that describes the differences in flow or speed at free flow and congestion conditions, which are exactly the types of conditions we need to predict. The second variation has a high frequency and can be removed by filtering (smoothing) the speed or flow signal. We tried two approaches to improving prediction. The first approach involves smoothing the data itself by using 15-minutes aggregated data instead of 5 to train and test the proposed PLSR. The second approach involved using the five minutes aggregated data to build the PLSR models and then smoothing the prediction result. In other words, we averaged the predicted result to get 15-minutes of aggregated prediction. Due to the limited space in this paper, in the following subsection, we will present only two figures showing the experimental results.

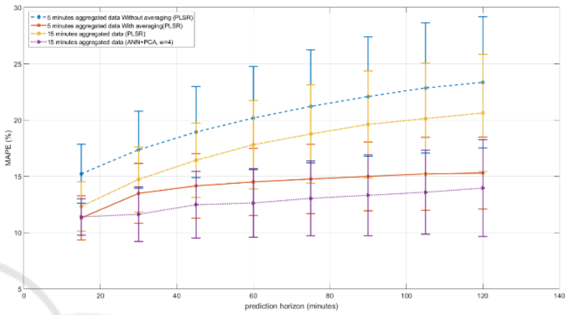

Figure 3: Comparison between the MAE (Vehicle per hour) of the Us-75 flow dataset.

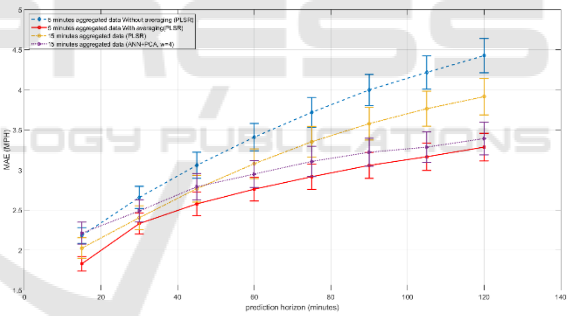

Figure 4: Graph Comparison between the MAE (MPH) of the Us-75 speed dataset.

The figures above show that smoothing the data by reducing its resolution to 15 minutes results in a lower MAE and MAPE rate compared to the 5 minutes aggregated data. This reduction in the errors is very good for the ANN+PCA case but not as good for the PLSR case. In the case of the second approach, which trains and the tests the PLSR models using five minutes aggregated data and then smooths the prediction results, the reductions in MAE and MAPE are good. .In conclusion, ANN+PCA gives a better result than PLSR when the training data is 15 minutes aggregated. If the data is 5 minutes aggregated, then using PLSR to build the models and smoothing the prediction result is recommended.

6 CONCLUSIONS AND FUTURE WORK

In this paper, two machine-learning techniques were used to predict the spatiotemporal evolution of traffic stream states. A divide and conquer approach was proposed to overcome the CPU computational and memory loads that occur for a large road stretch and large prediction horizons. The two techniques were compared by building prediction models for a 23.3-mile stretch of US-75. The models were compared using the MAE and the MAPE statistics. In order to reduce the training time needed for ANNs, PCA was used to reduce the problem dimensionality using 50% of the principle components to cover almost all the variance in the data.

A sensitivity analysis was conducted to identify the optimum window size in the divide and conquer technique using the PLSR & ANN+PCA approaches. The experimental results showed the best window size to be 32 segments for PLSR; and 4 segments for the ANN+PCA because it reduces the ANN overfitting problem. Data aggregated at 5-minute and 15-minute intervals were used and the experimental results show that the ANN+PCA performed better than the PLSR approach when the 15-minute data the PLSR performed better. In the case of 5-minute aggregated data training, the ANN approach was found to be time consuming, rendering it impractical.

We should mention that the models proposed in this paper do not consider the response of travellers if the agencies operating the network disseminate predicted traffic information were sent to them. One area for future work is studying the interaction of the informed traveller and including the travellers' response as an input factor to the prediction models. Another area for future work is network-wide traffic prediction, for which models to predict the traffic on different roadway segments in a network. For the network-wide prediction, we can make use of newly available technologies along with new big data techniques to integrate travel behaviour and enhance traffic predictions. Moreover, the traffic patterns inside cities are dynamic and change over time, so online learning algorithms that continue to learn from each test (unseen example) in order to capture the dynamics of the traffic patterns are also needed.

REFERENCES

Abdi, H. Partial Least Square Regression (Pls Regression).

Ahmed, M. S. & Cook, A. R. 1979. *Analysis Of Freeway Traffic Time-Series Data By Using Box-Jenkins Techniques*.

Davis, G. A. & Nihan, N. L. 1991. Nonparametric Regression And Short□Term Freeway Traffic Forecasting. *Journal Of Transportation Engineering*.

Lee, S. & Fambro, D. 1999. Application Of Subset Autoregressive Integrated Moving Average Model For Short-Term Freeway Traffic Volume Forecasting. *Transportation Research Record: Journal Of The Transportation Research Board*, 179-188.

Park, B., Messer, C. & Urbanik Ii, T. 1998. Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network. *Transportation Research Record: Journal Of The Transportation Research Board*, 39-47.

Sun, H., Liu, H., Xiao, H., He, R. & Ran, B. 2003. Use Of Local Linear Regression Model For Short-Term Traffic Forecasting. *Transportation Research Record: Journal Of The Transportation Research Board*, 143-150.

Van Der Voort, M., Dougherty, M. & Watson, S. 1996. Combining Kohonen Maps With Arima Time Series Models To Forecast Traffic Flow. *Transportation Research Part C: Emerging Technologies,* 4, 307-318.

Vlahogianni, E. I., Karlaftis, M. G. & Golias, J. C. 2014. Short-Term Traffic Forecasting: Where We Are And Where We're Going. *Transportation Research Part C: Emerging Technologies,* 43, Part 1, 3-19.

Young-Seon, J., Young-Ji, B., Mendonca Castro-Neto, M. & Easa, S. M. 2013. Supervised Weighting-Online Learning Algorithm For Short-Term Traffic Flow Prediction. *Intelligent Transportation Systems, Ieee Transactions On,* 14, 1700-1707.

Zheng, W., Lee, D.-H. & Shi, Q. 2006. Short-Term Freeway Traffic Flow Prediction: Bayesian Combined Neural Network Approach. *Journal Of Transportation Engineering,* 132, 114-121.