A Car-Following Model Using Online Sequential Extreme Learning Machine

X H Yan1*, Z W Zhou1

1School of Automation, Chongqing University, Chongqing, 400030, China
*Corresponding author’s e-mail: yanxiaohaoahh@gmail.com

Abstract. To prevent the sedimentation and loss of real-time data reflecting the car-following behavior and use it for modeling services, a car-following model using online sequential extreme learning machine is proposed. This model based on car-following behavior is affected by the driver’s memory effect, and a forgetting factor is introduced according to the timeliness of incremental data in different traffic conditions. It is trained, validated, and tested by NGSIM project data. Compared with other data-driven car-following models, the model has higher predictive accuracy and better real-time performance. With the help of experimental simulation, it verifies that the average time of the model’s online update is about 0.004s, which can fully meet the real-time requirements of the online traffic simulation system. The research results show that the car-following model proposed in this article is reasonable and feasible.

1. Introduction

The Car-following model (CFM) is an important part of the traffic simulation system. It is used to study the interaction between front and rear vehicles in a single lane, immutable lane, congestion, and other traffic conditions. With the development of intelligent transportation systems, intelligent devices provide huge amounts of real-time data for online traffic simulation systems. The predictive accuracy of the car-following model is not ideal, because of the limit by lack of offline data. In 2018, a car-following model based on online support vector regression was proposed in [1], which realized updating online. However, the essence of the model is to adjust the original training set by incremental data, and the update of the model still depends on the whole data set. Therefore, a car-following model is proposed which can be updated in real-time only with incremental data in this paper.

The car-following behavior is continuous in time, and the state of the next moment is related to the state of the historical moment. In 2003, it found that the driver’s memory has a vicious effect on the macro traffic flow [2]. In 2016, The research of [3] showed that the car-following model simulates the driving behavior better by considering the driver’s memory. To fully consider the memory effect of drivers, many researchers proposed car-following models based on the deep neural network [4-6]. However, due to the complex network structure of these models and the long training time, it is not possible to update the models with real-time incremental data. Therefore, these models have some limitations in the real-time online traffic simulation system.

To sum up, in order to effectively utilize real-time data reflecting the car-following behavior and based on the driver’s memory effect, a car-following model based on an online sequential extreme learning machine is established, trained, and tested. The model is compared with the car-following model based on Back Propagation Neural Network (BPNN) and Support Vector Regression (SVR). The results show that the model has high predictive accuracy and real-time performance, and the specific details are described later.
2. Establishment of car-Following Model

The theoretical-driven car-following models are difficult to use online incremental data and are not suitable for online traffic simulation system. The existing data-driven car-following models do not meet the real-time requirements of the online traffic simulation system, because of some drawbacks, such as slow convergence, difficult training, and easy to fall into local infinitesimal value. Therefore, this paper chooses the extreme learning machine with a shorter training time and strong generalization ability to build a car-following model suitable for online traffic simulation.

2.1. ELM

In 2006, Extreme Learning Machine (ELM) was proposed in [7]. The main feature of ELM is that the weights and bias from the input layer to the hidden layer can be determined randomly.

Suppose a training sample set consisting of \( N \) samples \( \{(x_1, t_1), \ldots, (x_n, t_n)\} \), where \( (x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m \), \( x_i = [x_{i1}, x_{i2}, \ldots, x_{im}]^T \in \mathbb{R}^n \) denotes the \( i \)th input feature vector, \( t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathbb{R}^m \) denotes the \( i \)th label vector. If the activation function from the input layer to the hidden layer is represented by \( g(\cdot) \), \( l \) is used to represent the number of hidden layer nodes, and \( \beta_i = [\beta_{i1}, \beta_{i2}, \ldots, \beta_{imi}]^T \in \mathbb{R}^m \) denotes the weight vector connecting the \( i \)th hidden node and the weight output nodes, \( w_i \) denotes input weight vector of the \( i \)th hidden layer node, \( b_i \) denotes input bias of the \( i \)th hidden layer node, \( w_i \cdot x_i \) denotes the inner product of product of \( w_i \) and \( x_i \), and ELM can be expressed as in equation (1).

\[
H\beta = T \tag{1}
\]

Where, \( H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_l \cdot x_1 + b_l) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_n + b_1) & \cdots & g(w_l \cdot x_n + b_l) \end{bmatrix}_{n \times l} \)

\( \beta = \begin{bmatrix} \beta_{11} \\ \vdots \\ \beta_{lm} \end{bmatrix} \)

\( T = \begin{bmatrix} t_{11} \\ \vdots \\ t_{nm} \end{bmatrix} \)

To improve the generalization ability of ELM, considering the empirical error minimization and structural risk minimization comprehensively, the optimization goal can be expressed in equation (2).

\[
\arg \min_\beta \frac{1}{2} ||\beta||^2 + \frac{c}{2} \sum_{i=1}^{n} ||\xi_i||^2 \\
\text{s. t.} \ h(x_i)\beta = t_i^T, i = 1, \ldots, n. \tag{2}
\]

Where \( c \) represents the punishment factor, \( \xi_i=[\xi_{i1}, \xi_{i2}, \ldots, \xi_{imi}] \) indicates the deviation between output vector and the label vector of the \( i \)th inputs, \( h(x_i) \) is the \( i \)th hidden node output vector for input \( h(x_i) \).

The optimal solution of equation (2) can be expressed in equation (3).

\[
\beta = \begin{cases} (H^TH + \lambda)^{-1}H^T T, & N > l \\ H^T (HH^T + \lambda)^{-1} T, & N < l \end{cases} \tag{3}
\]

Where \( \lambda = \frac{1}{c} \).

To facilitate incremental updating of the model, an Online-Sequential Extreme Learning Machine (OS-ELM) is proposed in [8]. In order to realize the sequential update of the weights, OS-ELM modified the weight update equation of ELM to equation (4) and equation (5) by the least square method.

\[
P_{k+1} = P_k + P_k H_{k+1}^T (I + H_{k+1} P_k H_{k+1}^T)^{-1} H_{k+1} P_k \tag{4}
\]

\[
\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta^{(k)}) \tag{5}
\]

Where \( P \) is the intermediate variable matrix of weight update and \( P_0=(H_0^T H_0)^{-1} \).

2.2. OS-ELMCF

Figure 1 shows the Online-Sequential Extreme Learning Machine Car Following (OS-ELMCF) model structure using OS-ELM. Different from the driver’s memory, the historical moment data is used as the...
model input to embed the driver’s memory effect, and the velocity of the following vehicle is used as the model output.

![Figure 1](image.jpg)

Figure 1. This is the structure diagram of the OS-ELMCF model. $X$ is the input features vector, $Y$ is the model output vector, $\beta$ is the hidden layer output weight matrix, $W$ and $B$ are the learning parameters of hidden nodes.

In the OS-ELMCF model, $V_f(i)$ is the velocity of the following vehicle at the $i$th moment, $V_p(i)$ is the velocity of the leading vehicle at the $i$th moment, $d(i)$ is the distance of two vehicles at the $i$th moment, $V$ is the longest predicted time, and $K$ is the historical time.

To prevent overfitting of the model, each group of car-following incremental data learning is completed, the performance of the model is evaluated by using a $Q$ group data set. If the performance of the car-following model is not better than before updating, the model reverts to the previous model. This test data set updates after several consecutive model update failures, and while adding new observation data, the oldest test sample data will be discarded in turn to keep the number of test samples.

All observations are weighted equally in OS-ELM, but in actual traffic roads, the formation and dissipation of car-following fleets are accompanied by changes in traffic conditions, and the impact of interference on car-following behavior is different under different traffic conditions. A forgetting factor was introduced in OS-ELM to improve the timeliness of training in [9]. Inspired by this idea, the OS-ELMCF model introduces a learning rate factor related to the velocity of car-following. After introducing the learning rate factor, the OS-ELMCF can be expressed in equations (6) and (7).

$$
P_{k+1}^{-1} = r P_k^{-1} + H_{k+1}^T H_{k+1}$$

$$
\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta^{(k)})$$

Where $r = \tanh (2.3 + \frac{V_d}{V})$ represents the model learning factor, $V_d$ is the velocity of the vehicle which follows at low speed, the value usually is 25 ft/s, and $P_0^{-1} = H_0^T H_0$, $V$ is the measured velocity of the following vehicle. The construction of OS-ELMCF is divided into two stages as follows: offline stage and online stage. The offline stage is summarized as algorithm 1-1, while the online stage is summarized as algorithm 1-2.

---

**Algorithm 1-1 offline model building algorithm**

**Input:**
- $\{x_i, t_{i} \}_{i=1}^{N}$: offline car-following data set;
- $L$: number of neurons in the hidden layer;
- $g(\cdot)$: activation function;

**Output:**
- $\beta^{(0)}$: outputs weight matrix of the offline car-following model;
- $P_0 = (H_0^T H_0)^{-1}$, the intermediate variable matrix of weight update;

1: randomizing the learning parameters $W$ and $B$ of hidden nodes;
2: calculating the output matrix of hidden layer $H_0$;
3: $\beta^{(0)}$ is calculated by equation (3) and $P_0$ is calculated according to $H_0$;
4: return $\beta^{(0)}, P_0$.

Algorithm 1-2 online model updating algorithm

Input:
$L, g(\cdot), \beta_0$ and $P_0$;
Output:
$\beta^{(i)}$: the $i$th output weight matrix of online model;
$P_i$: the $i$th the intermediate variable matrix of weight update;
1: while online simulation do
2: while there is car-following phenomenon do
3: updating $P_i$ according to equation (6) and updating $\beta^{(i)}$ by equation (7);
4: predicting $V_{(t+j\Delta t)}$ by the updated model, $j=1, ..., m$;
5: end while
6: if the updated model has better generalization ability on the $Q$ test set:
7: offline model update;
8: end while
9: return $\beta^{(i)}, P_i$.

3. Data analysis and processing

3.1. Data preparation
NGSIM project data set is high-precision vehicle trajectory data produced by federal highway administration in December 2003, April 2005, and June 2005. This data set provides a unified platform for model research and parameter calibration for researchers. Because the vehicle trajectory data in the NGSIM data set has high accuracy and various data types, it can fully reflect the actual car-following behavior. Therefore, this paper selects the vehicle trajectory of the US-101 road section in June 2005 during 08:05-08:20 is selected as the data. To obtain the accurate car-following trajectory data of this data set, the vehicle trajectory data is processed and filtered according to the following conditions:

(1) Due to the existence of abnormal data and noise data in the raw data, the velocity of the vehicle and space to vehicle ahead are processed by the symmetric exponential smoothing method in [10].

(2) To ensure reliable and sufficient vehicle trajectory data during the car-following process, the car-following time is at least 40s, and the space between the two vehicles is not more than 50m.

(3) Considering that the driver's reaction time is generally between 0.4s to 2.3s in [11], this paper selects the car-following vehicle trajectory data with a period of 1s.

A car-following data set which contains 986 groups of vehicles is obtained after processing and filtering, it contains 64856 pieces of data.
3.2. Uncertainty of car-following behavior under different traffic conditions

(a) The relationship between velocity of following vehicles and space of car-following pairs

(b) The relationship between velocity difference and space of car-following pairs

Figure 2. The relationship diagram of variables in car-following data

Figure 2 shows the relationship of variables in the selected car-following data. In figure 2 (a), it can be seen that there is a positive correlation between the velocity of the following vehicles and the space to vehicle ahead, and the car-following behavior has higher uncertainty when the road is smooth. Figure 2 (b) shows that the velocity difference of car-following behavior is basically distributed around 0 ft/s, and most of the car-following data are disturbed, which is consistent with the reality. Therefore, the selected data set is suitable for testing the performance of the proposed model, which can apply incremental data.

3.3. Selection of offline model parameters

The performance index selects the velocity of the following vehicle at the next moment, and the evaluation method uses Root Mean Square Error (RMSE). Equation (8) represents the specific expression of RMSE.

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\]

Where \( n \) is the total number of the test set, \( y_i \) represents the \( i \)th measured value, and \( \hat{y}_i \) represents the \( i \)th predicted value.

The model has 4 undetermined parameters, which are the number of hidden layer neurons \( L \), the \( \lambda \) value, the number of historical data moments \( K \), and the number of predicted moments \( V \). Since the performance index, \( V \) is taken as 1 in this paper. The study [11] has shown that \( \lambda \) is bounded between 0 and 1. To simplify the parameter calibration process, the value of \( \lambda \) selects 0.8 and the value of \( L \) selects 100, 200, 300, 400, and 500 respectively.

According to the time of the data set, 250 groups of car-following data are selected as the training set, which contains the car-following data when the road is unobstructed and congested. Selecting randomly two groups of car-following data under different traffic conditions as the test set. Figure 3(a) shows the impact of data at each historical time during congestion on the prediction. When BPNN and ELM consider two historical time data in figure 3(a), the accuracy is significantly improved, and the final accuracy tends to stabilize. And SVR stabilized after considering 4 historical moment data. At the same time, the predictive accuracy of ELM stabilizes after the number of hidden layer neurons increases to 300. Figure 3(b) shows the impact of data at each historical time during unobstructed periods on the prediction. Compared with congestion, the predictive accuracy is improved during the unobstructed time, and it is not a greater predictive performance while considering historical time during unobstructed periods.

To sum up, the model selection parameters combination is expressed in equation (9).
\[
\begin{align*}
V &= 1 \\
L &= 300 \\
\lambda &= 0.8 \\
K &= 4
\end{align*}
\]  
(9)

(a) The impact of data at each historical moment on the forecast during congestion  
(b) The impact of data at each historical moment on the forecast during unobstructed periods

**Figure 3. The impact of historical data on forecasts**

4. Comparative verification and experimental analysis

In order to test the performance of OS-ELMCF, it is compared with the car-following model based on SVR and BPNN. The models all use the four historical velocities and space between the two vehicles as the input and the next moment velocity of the following vehicle as the output. SVR parameters are set according to the literature [12], and BPNN parameters are set according to the literature [13]. Experimental operating environment, processor: Intel(R) Core(TM) i5-3210M CPU @ 2.50GHZ, memory size: 8.00GB, Matlab 2017a.

Using 600 groups of car-following data as the training set, it contains 40200 pieces of data. Selecting randomly two groups of car-following data under different traffic conditions as the test set. To test the online learning ability of the proposed model, ten groups of car-following data were selected before the test groups. The velocity of the following vehicle is selected as the performance index, and \( R^2 \), the Mean Absolute Error (MAE), the coefficient of determination (\( R^2 \)), and the Theil’s inequality coefficient (\( U \)) as the evaluation method. Among them, MAE represents the average value of the absolute value of the deviation between the predicted value and the measured value, which is represented in equation (10). \( R^2 \) describes the correlation between the predicted value and the measured value, and it is represented in equation (11). The numerator of the coefficient of Theil's inequality is the root mean square error, and the denominator is the sum of the mean square of the observed value and the predicted value, which can represent the relative root mean square error, expressed in equation (12).

\[
MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| 
\]  
(10)

\[
R^2 = \frac{(\sum_{i=1}^{n} y_i \sum_{i=1}^{n} \hat{y}_i - n \sum_{i=1}^{n} y_i \bar{\hat{y}})^2}{(n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2) \times (n \sum_{i=1}^{n} \hat{y}_i^2 - (\sum_{i=1}^{n} \hat{y}_i)^2)} 
\]  
(11)

\[
U = \left( \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \right)^{1/2} / \left( \frac{1}{n} \sum_{i=1}^{n} y_i^2 \right)^{1/2} + \left( \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i^2 \right)^{1/2} 
\]  
(12)

In these formulae, \( n \) is the total number of test samples. The smaller the \( MAE \) value and the \( U \) value are, the closer the predicted value is to the measured value. The value range of \( R^2 \) is \([0, 1]\), the more the value tends to 1, the better the model fitting effect, on the contrary, the worse the fitting effect.

Figure 4 shows the comparisons of prediction results of SVR, BPNN, offline model, and online model. Notably, the elliptical area is partially enlarged in figure 4. It can be seen from (a) and (b) in figure 4 that the proposed model has more accurate prediction accuracy. In general, the prediction accuracy of SVR is the worst and the OS-ELMCF is the best. There is little predictive difference between
BP, SVR, ELM, and the OS-ELMCF at the beginning of car-following in figure 4(c) and figure 4(d). With the increase of prediction time, it is obvious that OS-ELMCF can predict more accurately due to the use of incremental real-time data.

Figure 4. Comparisons of car-following behavior prediction between SVR, BP and the proposed model

Table 1 and table 2 show the evaluation results of two groups of car-following respectively. The results show that the OS-ELMCF model is the closest to measured values in different traffic conditions. Obviously, the performance of OS-ELMCF is better than that of ELM, indicating that OS-ELMCF has the online learning ability for incremental data. At the same time, the training of the proposed offline model is 4 times faster than SVR and nearly 20 times faster than BP. Especially, the time of OS-ELMCF model update is as low as 0.004s, which fully meets the real-time requirements of online traffic simulation.

Table 1. Evaluation table of car-following results of vehicle No. 1216 (unblocked)

| model    | MAE  | RMSE | $R^2$ | $U$  | Training /online update duration (s) |
|----------|------|------|-------|------|--------------------------------------|
| BP       | 0.695| 0.834| 0.959 | 0.008| 19.043                                |
| SVM      | 1.748| 1.967| 0.925 | 0.045| 4.0135                                |
| ELM      | 0.621| 0.724| 0.958 | 0.006| 0.9325                                |
| OS-ELMCF | **0.564** | **0.672** | 0.958 | **0.005** | **0.9325(0.004)**                  |

Table 2. Evaluation table of car-following results of vehicle No. 1619 (congestion)

| model    | MAE  | RMSE | $R^2$ | $U$  | Training /online update duration (s) |
|----------|------|------|-------|------|--------------------------------------|
| BP       | 0.798| 2.907| 0.926 | 0.194| 19.043                                |
| SVM      | 3.624| 4.644| 0.926 | 0.472| 4.0135                                |
| ELM      | 0.796| 2.907| 0.926 | 0.194| **0.9325**                            |
| OS-ELMCF | **0.762** | **2.897** | **0.927** | **0.193** | **0.9325(0.004)**                   |

To sum up, the OS-ELMCF model has the ability of online real-time update and higher predictive accuracy. However, single car-following data is too few, the car-following data is about 35s when the road is unblocked and the car-following data is about 90s when the road is congested. Due to the lack of car-following data of a single vehicle, the performance improvement of the OS-ELMCF model is not
great. The advantage of the OS-ELMCF model will be reflected after learning the incremental data in the online traffic simulation process.

5. Conclusion
Under different traffic conditions, the OS-ELMCF model prediction results are the closest to the measured values. OS-ELMCF model can learn one or a group of incremental data online, and the average online update time is about 0.004 s, which fully meets the real-time requirements of an online traffic simulation system. Therefore, the OS-ELMCF model is reasonable and feasible. However, the prediction performance of the model in different areas, different traffic conditions, different driving modes, and other scenarios needs to be further studied.

Acknowledgments
This work is supported by the Technology Innovation and Application Development Project of Chongqing (CSTC2019jscx-fxydX0026).

References
[1] H. Wang and F. Chen. (2018) A car-following model based on online support vector regression. Electronic Technology, 47(11): 21-28.
[2] H. M. Zhang. (2003) Driver memory, traffic viscosity and a viscous vehicular traffic flow model. Transportation Research Part B Methodological, 37(1): 27-41.
[3] Pei, Xin, Wang, Haixin, Wong, S., C., Choi, Keechoo and Pan. (2016) Empirical evidence and stability analysis of the linear car-following model with gamma-distributed memory effect. Physica A Statistical Mechanics & Its Applications.
[4] X. Huang, J. Sun and J. Sun. (2018) A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. Transportation Research Part C: Emerging Technologies, 95(OCT.): 346-362.
[5] X. Wang, R. Jiang, L. Li, Y. Lin, X. Zheng and F.-Y. Wang. (2018) Capturing car-following behaviors by deep learning. IEEE Transactions on Intelligent Transportation Systems, 19(3): 910-920.
[6] L. Ma and S. Qu. (2020) A sequence to sequence learning based car-following model for multi-step predictions considering reaction delay. Transportation Research Part C: Emerging Technologies, 120: 102785.
[7] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew. (2006) Extreme learning machine: Theory and applications. Neurocomputing, 70(1-3): 489-501.
[8] N. Y. Liang, G. B. Huang, P. Saratchandran and N. Sundararajan. (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw, 17(6): 1411-23.
[9] H. Zhang, S. Zhang and Y. Yin. (2017) Online sequential ELM algorithm with forgetting factor for real applications. Neurocomputing, 261: 144-152.
[10] C. Thiemann, M. Treiber and A. Kesting. (2008) Estimating Acceleration and lane-changing dynamics from next generation simulation trajectory data. Transportation Research Record: Journal of the Transportation Research Board, 2088(1): 90-101.
[11] D. Xiao-Jian, L. Yuan, Z. Zhi-Feng and X. Xin. (2017) Optimization extreme learning machine with v regularization. Neurocomputing: S0925231217301996.
[12] B. Lu. (2017) Modeling and analysis of car-following behavior using data-driven methods.
[13] C. Colombaroni and G. Fusco. (2014) Artificial neural network models for car following: Experimental analysis and calibration issues. Journal of Intelligent Transportation Systems, 18(1): 5-16.