Multi Long-Short Term Memory Models for Short Term Traffic Flow Prediction

SUMMARY Many single model methods have been applied to real-time short-term traffic flow prediction. However, since traffic flow data is mixed with a variety of ingredients, the performance of single model is limited. Therefore, we proposed Multi-Long-Short Term Memory Models, which improved traffic flow prediction accuracy comparing with state-of-the-art models.

key words: multi-lstm models, k-means clustering, short term traffic flow prediction

1. Introduction

In the past few decades, cars have gradually become one of the major tools of transport, which brings traffic congestion on urban freeways in many countries. Compared with the blind expansion of traffic capacity which consumes amounts of resources, the construction of intelligent transport system (ITS) [1] is a faster and more effective method to ease traffic pressure, which aims to make full use of existing urban road network. In fields of ITS research, short term traffic flow prediction (STTFP) is one of the hottest topics [2] and it acts as an important basis for guiding traffic flow and driving path planning.

As we know, many researchers have focused on short term (about 5 to 30 minutes) traffic prediction, which preferred to utilize linear and parametric methods in early research on STTFP. One of the most widely used methods is autoregressive integrated moving average (ARIMA) model. A hybrid model combining linear ARIMA model with nonlinear GARCH model, was proposed by Chen et al. [3] to capture both the conditional mean and conditional heteroscedasticity of traffic flow series. Kumar and Vanajakshi [4] proposed seasonal ARIMA model to predict traffic flow with limited input data. However, traffic condition data is characteristically stochastic rather than chaotic, which has been proved in [5]. Due to the stochasticity and nonlinearity of the traffic flow sequence, researchers paid more attention to nonparametric or nonlinear method. Smith and Demetsky [6] compared several models for traffic flow prediction and revealed that k-nearest neighbors (KNN) model is superior to linear ARIMA and historical average (HA). Moreover, Zhang et al. [7] analyzed the parameters of both average and weighted KNN regression model to optimize prediction effect. Another nonlinear and supervised statistical method is support vector regression (SVR). The basic idea is to map the input traffic flow data into the high-dimensional feature space and to find the linearly divided hyper plane in the space. Hu et al. [8] proposed a hybrid PSO-SVR forecasting method to search optimal SVR parameters with less learning time and deserve accurate forecasting when data contains noises. Recently, deep learning approaches are also successful in traffic flow prediction, especially, long short term memory (LSTM) network performed well in time series prediction [9].

The methods mentioned above are all based on a single model, but traffic flow is a time series that contains a variety of ingredients, so the performance of a single model may be limited. Because when a model has a good predictive effect for time series on heavy load, it’s difficult to ensure that it has the same effect on light load [10]. What’s more, Qi et al. [11] proposed that congestion and non-congested traffic flow should be modeled separately. In fact, after observing and analyzing the traffic flow waves, we found two interesting phenomena. Firstly, the first-order difference of time series can cause an upward trend or a downward trend of waves. Secondly, taking absolute value of the difference can cause a violent trend or a gentle trend. In order to distinguish traffic flow under different situations and solve the defect of single model method, we firstly extracted a feature which characterizes different trends of traffic flow, then divided the flow into two categories and modeled them separately.

As for training and prediction, we proposed a multi-model method based on LSTM. The reason is that long-short term memory networks are good at processing sequential data [9]. Compared to single model based methods, our method can handle the complicacy and diversity of traffic flow better, by analyzing and predicting different flow distributions respectively. Moreover, our proposed framework can be easily applied to online scenario.

2. The Proposed Prediction Architecture

As shown in Fig. 1, traffic flow data, which is defined as the number of vehicles passing an observation area in a period of time, is divided into training set and testing set. Training data are further clustered as gentle trend and violent trend, which are used to train two sub-models adapting to two traffic condition. Testing data need to be classified and predicted by corresponding LSTM sub-model.
2.1 Data Pre-Processing

Traffic flow data are aggregated with different time interval to meet the needs of different application scenarios. Then the data need to be normalized in order to reduce the effects of peak differences in traffic flow every day. The normalized way is shown in (1), where \( \{x_t\} \) is a sequence of traffic flow data. The last step is to select the appropriate length \( N \) of a sliding time window.

\[
y_t = \frac{x_t - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{1}
\]

Traffic flow forecasting can be described as predicting traffic flow data at time \( N+1 \) based on a sliding time window sequence \( \{x_t\}_{t=1}^{N} \). Therefore, the last step is to select the appropriate length \( N \) of a sliding time window.

2.2 Extracting Feature and Clustering

Since the proposed method is intended to establish corresponding models for different categories of data, it is important to separate traffic flow data into appropriate categories. Through lots of observations and analysis of traffic flow waves, we determined to extract meaningful features for each sample and apply K-Means clustering.

At the beginning, we tried to use only the first-order difference as the feature of trends, for example, a sample like \( \{x_1, x_2, \ldots, x_N\} \) can provide a feature vector \( \{x_2 - x_1, x_3 - x_2, \ldots, x_N - x_{N-1}\} \) of \( N-1 \) dimensions. If \( x_t - x_{t-1} \) is larger than zero it represents an upward trend, otherwise a downward trend. Then we used the Euclidean metric to measure distance between features and clustered them by K-Means (\( K=2 \)). But the clustering effect of this scheme is not satisfying. To visualize the clustering results, we used PCA to reduce \( N-1 \) dimensions to 2, as shown in Fig. 2(a). It seems that two categories of samples are mixed, which indicates that such feature is not appropriate and lack of distinguishability. Then through observing and experiment, we find that taking the absolute value after first-order difference like \( \{|x_2-x_1|, |x_3-x_2|, \ldots, |x_N-x_{N-1}|\} \) can clearly divide traffic flow samples into two categories, as shown in Fig. 2(b).

One kind of samples changes slowly, another rapidly, and we respectively name them the gentle trend and the violent trend. The length \( N \) of sliding window is 6 and the time interval is 5 minutes here. When \( N \) changes from 2 to 12, the effect of such clustering method is almost the same as that shown Fig. 2(b). The only difference is the quantity of two categories of samples, but it has little effect on the final prediction accuracy.

In order to present the clustering effect intuitively, we pick out a whole day traffic flow and framed the gentle samples with green rectangles and the violent samples with red ones. As shown in Fig. 3, the violent samples either change rapidly between the beginning and end or fluctuates greatly itself, while gentle samples behave the opposite.

2.3 Multi-LSTM Models Training

In traditional recurrent neural network (RNN), during the back propagation through time (BPTT) process, the gradient signal propagates along the hidden layer and multiply the weight matrix of the neurons, which can lead to the end of learning process if the gradient signal tends to either blow up or vanish. Long Short Term Memory (LSTM) introduces a new structure called a memory cell to solve this problem. A memory cell is composed of four main elements, which are known as input gate, neuron with self-recurrent connec-
tion, forget gate and output gate. In this paper, we propose a new training method, which is the key idea in the architecture of multi-model LSTM. At first, all training data are used to train a master LSTM model as shown in Fig. 1. The architecture of the LSTM consists of an input layer, a hidden layer with LSTM blocks, a mean-pooling layer and an output layer. The input neuron number equals to sliding time window length $N$ and the size of output layer is 1. The number of LSTM blocks in LSTM layer is set in range from 20 to 100 with a step of 20. The purpose of this step is to generate model parameters, which will be used as initialization parameters for sub-model training. The significance of the master model is to capture the overall trend of traffic flow data, which is conducive to enhance generalization ability of sub-models and prevent them from failing into local optimal solution.

Initialized by model parameters of master model, two LSTM sub-models are trained separately by two types of traffic flow data called violent and gentle trend, which are divided by K-Means clustering. At last, we acquire two sub-models, respectively prepared for forecasting two kinds of traffic situation.

2.4 Online Forecasting Program

Since the category of samples need to be judged before using corresponding sub-model to predict in the real application scenarios, before prediction we added a K-Nearest Neighbor (KNN) classifier, which is trained with two types of data generated by K-Means clustering. Through a large number of experiments, we set the $K$ of KNN as 10 and assigned the voting weight to the reciprocal of the Euclidean distance to achieve higher classification accuracy. After classifying by KNN, the sample can be sent to corresponding sub-models.

3. Experiments and Results

3.1 Details of Database

The experimental data applied on our proposed model are from the Caltrans Performance Measurement System (PeMS), which has been collecting historical traffic data every 30s from major cities in California for more than ten years. The system aggregates the data into a minimum interval of 5 minutes for each vehicle detection station (VDS), which can be retrieved and downloaded on its official website. The new version of the system completes the missing data caused by a sensor abnormality and the data quality can reach 99.8 percent observation. In our study we chose the No.716921 VDS, it contains complete traffic flow data of 52 weeks collected from Jan 1st to Dec 30th in 2015. We only paid attention to traffic flow prediction on weekdays, which is more meaningful to our lives [9], [11]. Therefore, we removed the data of weekends and holidays, and there still remains 247 weekdays in 2015. We picked the first 200 days (about 80 percent of the whole data) for training, while the rest 47 days for testing.

3.2 Evaluation Criterion

In our study, we used two common evaluation criteria to evaluate our models, they are Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE), which are shown in Eq. (2) and (3).

$$\text{MAPE} = \frac{1}{M} \sum_{i=1}^{M} \frac{|x_i - \hat{x}_i|}{x_i}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{M} (x_i - \hat{x}_i)^2}{M}}$$

$M$ is the size of testing set, $x_i$ is the predicted value of the $i^{th}$ test sample and $\hat{x}_i$ is the real traffic flow data of $i^{th}$ period called real label corresponding to the $i^{th}$ test sample. By calculating such absolute and relative differences between many predicted values and real values, we assess whether our method achieves the predicted goal.

3.3 Comparison of Prediction Models

To evaluate the effectiveness of the proposed multi-model LSTM (M-LSTMs), we compared our method to four representative methods, which are historical average (HA), k-nearest neighbors (KNN), support vector regression (SVR) and long-short term memory network. We call the fourth model the single-model LSTM (S-LSTM) to distinguish from our method. We also designed two contrast experiments to investigate the optimal length $N$ of time window for each method and then evaluate their performance under different time intervals.

Through the grid search, we determined the optimal parameters of each method in our dataset. HA model was a baseline here. It used an average of past traffic flow to predict future traffic flow. For KNN, we set the parameter $K$ as 25 and assigned the same weights for all points in each neighborhood. In SVR, we set the penalty parameter $C$ as 2.0 and used ‘rbf’ as kernel. The hidden layer of S-LSTM contains 100 LSTM blocks, which is the same to our proposed model.

The first contrast experiment is an investigation on the optimal length $N$ of sliding time window for each method. The $N$ ranges from 2 to 12 and the time interval is 5 minutes. We can safely draw two conclusions from Fig. 4 (a). Firstly, the performance of our model is superior to the others for different length of time window. Secondly, the performance of KNN and SVR are more affected by sliding time window, while S-LSTM and M-LSTMs are hardly affected. The optimal time window length $N$ of them are 9 for KNN, 11 for SVR, 6 for S-LSTM and 11 for M-LSTMs. In order to compare two kinds of LSTM model under the same settings, we also assigned time window length $N=6$ for M-LSTMs in next contrast experiment. The second experiment is the performance evaluation of 5 models under different time intervals. The time interval was set as 5-min, 10-min, 15-min and
20-min, respectively. As mentioned above, we used MAPE and RMSE to measure the accuracy of the forecasting. As the time interval increases, the label $x_t$ of each traffic flow sample also increases. From the definition we know that the size of label has no effect on MAPE. But the RMSE increases as the size of label increases. Therefore, in general MAPE is the primary indicator to evaluate the performance of a model for different time intervals. Smaller MAPE indicates a better forecasting model. On the other hand, RMSE is effective only when it is used to evaluate different models under the same time interval.

As shown in Table 1, we can find that MAPE and RMSE of the M-LSTMs are both lower than that of the other models, which proves the validity of our model. What’s more, when time interval is 15-min, MAPE of all models except HA reached the minimum, which may reveal that 15-min is the most suitable interval in practice.

According to the results of two experiments above, we randomly took one day to compare real traffic flow values and the predicted values using M-LSTMs, where the length of sliding time window is 6 and time interval is 15-min. As shown in Fig. 4 (b), they are quite similar.

<table>
<thead>
<tr>
<th>Models</th>
<th>5-min</th>
<th>10-min</th>
<th>15-min</th>
<th>20-min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAPE(%)</td>
<td>RMSE</td>
<td>MAPE(%)</td>
<td>RMSE</td>
</tr>
<tr>
<td>HA</td>
<td>13.06</td>
<td>48.25</td>
<td>10.39</td>
<td>89.69</td>
</tr>
<tr>
<td>KNN</td>
<td>10.24</td>
<td>43.47</td>
<td>9.16</td>
<td>92.58</td>
</tr>
<tr>
<td>SVR</td>
<td>12.09</td>
<td>35.91</td>
<td>8.81</td>
<td>67.89</td>
</tr>
<tr>
<td>S-LSTMs</td>
<td>9.40</td>
<td>33.90</td>
<td>8.02</td>
<td>63.23</td>
</tr>
<tr>
<td>M-LSTMs</td>
<td>9.19</td>
<td>32.57</td>
<td>6.94</td>
<td>60.63</td>
</tr>
</tbody>
</table>

20-min, respectively. As mentioned above, we used MAPE and RMSE to measure the accuracy of the forecasting. As the time interval increases, the label $x_t$ of each traffic flow sample also increases. From the definition we know that the size of label has no effect on MAPE. But the RMSE increases as the size of label increases. Therefore, in general MAPE is the primary indicator to evaluate the performance of a model for different time intervals. Smaller MAPE indicates a better forecasting model. On the other hand, RMSE is effective only when it is used to evaluate different models under the same time interval.

As shown in Table 1, we can find that MAPE and RMSE of the M-LSTMs are both lower than that of the other models, which proves the validity of our model. What’s more, when time interval is 15-min, MAPE of all models except HA reached the minimum, which may reveal that 15-min is the most suitable interval in practice.

According to the results of two experiments above, we randomly took one day to compare real traffic flow values and the predicted values using M-LSTMs, where the length of sliding time window is 6 and time interval is 15-min. As shown in Fig. 4 (b), they are quite similar.

4. Conclusion

In this paper, we propose a multi-model LSTM Network for short term traffic flow prediction. Unlike the previous methods based on a single model, our method divides traffic flow data into two different trends and analyzes them respectively with sub-models. We compared our method with HA, KNN, SVR and single LSTM model on PeMS dataset. The results show that the proposed method is superior to other four competing methods. Since the extracted features are only suitable for classifying traffic flow data into two categories, only two LSTM sub-models are adopted in our method. For future work, it would be interesting to discover other methods for traffic flow data classification. More detailed classification and more LSTM sub-models may achieve better prediction effect. Furthermore, our model mainly considers temporal impact on traffic flow without the spatial impact from neighbor observation stations. Taking such factors into account may improve the performance.

Acknowledgments

This work was supported in part by National Natural Science Funds (61771199), Guangdong Science and Technology Research Plan (2015B010131004,2017A10101027), and Fundamental Research Funds for Central University (D2171560). We would like to appreciate the data resources from Caltrans Performance Measurement System.

References