Using an Ensembled Boosted Model for IoT Time Series Regression

Shuai Lin[1](✉), Kun Zhang[1], Renkang Geng[2], and Liyao Ma[2]

[1] Shandong Non-Metallic Materials Institute, Jinan 250031, Shandong, China
i531inshuai@126.com

[2] School of Electrical Engineering, University of Jinan, Jinan 250022, Shandong, China
202021200778@mail.ujn.edu.cn, cse_sunb@ujn.edu.cn

Abstract. As a typical regression model, time series prediction is a very important part of machine learning. With the development of urban roads and the increase of car ownership, traffic data is more closely related to machine learning. As a common time-series data in our life, traffic flow data has great research value and a wide range of application fields. Compared with the general time series, traffic flow data has larger data volume, stronger volatility, and higher requirements for accuracy and speed of prediction, but traditional algorithms often fail to achieve these goals. With the development of ensemble algorithm, it has outstanding performance in the field of classification and regression. Therefore, we choose XGB algorithm as the core algorithm of this experiment. In this paper, we introduce the working principle of the XGBoost algorithm, the acquisition of the traffic flow data used in the experiment, and the feature extraction of the traffic flow data in detail. Finally, we use the XGBoost algorithm to model and output the prediction results. In addition, we modified some very important parameters in XGBoost, such as iteration model, iteration number, etc., to explore the influence of each parameter on prediction accuracy when the XGBoost algorithm is used to predict traffic flow data.

Keywords: Traffic flow data · IoT Time Series · Ensemble learning · XGBoost

1 Introduction

Since entering the 21st century, with the development of science and technology and civilization, the quickening pace of life, we follow the travel demand is becoming bigger and bigger [1], strong ability of infrastructure in China has brought the rapid development of urban road, greatly convenient for people to travel, travel choice, as the most convenient and flexible car became the most used means of travel, The number of motor vehicles in China has been increasing year after year, and electric vehicles have been developing rapidly [2]. However, the explosion of the number of cars has also brought

This work is supported by Shandong Key R&D Program grant 2019JZZY021005.

S. Wang et al. (Eds.): IoTCare 2021, LNICST 415, pp. 368–375, 2022.
https://doi.org/10.1007/978-3-030-94182-6_28

great challenges to urban road traffic, leading to traffic jams, accidents and other abnormal situations, which has brought great inconvenience and risks to people's lives [3]. Therefore, the management of traffic congestion has become one of the major social problems in the 21st century. If the traffic flow data can be accurately predicted, the traffic congestion can be effectively controlled, and the traffic pressure can be alleviated [5].

The traffic flow data itself has a certain trend and regularity, but due to the complexity of time and the heterogeneity of space, the data itself also has strong volatility in a short time, especially when some abnormal conditions occur on the road, the traffic flow data may change abruptly [8]. With the development of machine learning, more and more machine learning algorithms have been used in the prediction of time series data. Traffic flow data, as a typical representative of time series, has great research value and use-value [9].

There are many algorithms that are good for machine learning to predict traffic flow data, including statistical model, non-parametric regression model, neural network model and support vector machine model [10]. However, due to the high demand for traffic data on the prediction speed and accuracy of the algorithm, the traditional modelling methods have certain limitations [12]. Therefore, in recent years, more and more integration algorithms have been used in the field of time series prediction, such as Adaboost (Adaptive Boosting) [13], GBDT (Gradient Boosting Decision Tree) [14] etc. In 2016, Tianqi Chen has published a new algorithm, XGBoost (Extreme Gradient Boost) algorithm [15] was proposed, which was optimized on the basis of GBDT, and greatly improved the speed and efficiency of computing, which not only guaranteed.

In this paper, we extract the features of the traffic flow data and then forecast the traffic flow data based on the modelling of the XGBoost algorithm. Through the prediction results, we judge the prediction accuracy of XGBoost for such data and discuss the influence of different parameters in XGBoost on the prediction accuracy. Finding out the parameter configuration that XGBoost algorithm is suitable for predicting traffic flow data.

2 Related Work

Ensemble Learning [16] is an algorithm that builds multiple learners and then combines them together in a certain way, which is mainly divided into three categories, Bagging, Boosting and Stacking. Here we mainly introduce Bagging and Boosting.

Bagging uses a sampling method with putting back to generate training data. It randomly sampled the initial training set through multiple rounds of putting back, and multiple training sets were generated in parallel, which corresponded to the training of multiple base learners (there is no strong dependency relationship between them). Then, these base learners were combined to construct a strong learner. By increasing the randomness of samples, the variance can be reduced.

In essence, the XGBoost algorithm is a method based on Tree structure combined with ensemble learning, whose basic Tree structure is CART (Classification and Regression Tree) [17]. A CART regression tree is assumed to be a binary tree, and the sample space is divided by constantly splitting features. The core idea is that in the input space where the training data set is located, each region is recursively divided into two subregions and the output value of each sub-region is determined. The criteria for delimiting molecular regions depend on the type of tree, and the square error minimization criterion is usually used for regression trees. The objective function generated by the CART regression tree is:

$$\sum\nolimits_{x_i \in R_m} (y_i - f(x_i))^2$$

The XGBoost model is also a special GBDT. The idea is to grow a tree by repeated feature splitting, adding a tree at a time (in effect, learning a new function) to fit the residual of the previous prediction. XGBoost can customize a set of objective functions, with the help of Taylor expansion (you only need to know the first and second derivatives of the loss function to find the loss function) into a quadratic function of one variable, to get the extreme point and extreme value. The target function of XGBoost is as follows:

$$L(\phi) = \sum\nolimits_i l\left(y_i, \hat{y}_i\right) + \sum\nolimits_k \Omega(f_k)$$

GBDT algorithm only uses first-order Taylor expansion for the loss function, while XGBoost uses second-order Taylor expansion for the loss function. On the one hand, the second derivative expansion of the XGBoost loss function can increase the accuracy, and on the other hand, it can customize the loss function, because the second derivative Taylor expansion can approximate a large number of loss functions more accurately. On the basis of GBDT, XGBoost improves the computing speed, accuracy and optimizes some functions. For example, for samples with missing eigenvectors, the sparse perception algorithm adopted by XGBoost can automatically learn its splitting direction.

In order to effectively prevent the algorithm from over-fitting, there are two main solutions in the field of machine learning. One is to reduce the number of characteristic variables we select, and the other is regularization [18]. Regularization measures the smoothness of our fitting curve and is usually used as a penalty term in the loss function. In machine learning, the objective function of most algorithms is composed of the loss function and additional terms, such as the objective function of XGBoost. This extra term is the regularization term. Intuitive understanding from the polynomial model, the arbitrary function can use polynomial fitting, to help our fitting curve is smooth, is each different power coefficient of x before, a zero (or very small value) the situation is, the more that each x, in the final fitting polynomial to effectively distinguish between different weights.

$$\mathrm{f}(x_i) = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 + \ldots\cdots + w_nx_n$$

Regularization is to get rid of the items with low weight and keep the items with high weight, thus reducing the order of magnitude of the eigenvector. Regularization has the advantage of our retained all the characteristics of the variables, and not to deleting feature variables, but to distinguish the importance of the feature vector, because, in the actual problem, each variable can be more or less a little influence on the prediction of every variable has the value of it, we don't want to remove them [20], However, we also hope that some characteristic variables will not excessively affect the curve to prevent over-fitting, so we adopt the regularization method.

3 Methods

The traffic flow data used in this experiment is typical time-series data. Specific acquisition and processing methods are as follows: at a node on a certain road, read the traffic flow of this section within 5 min at a fixed time interval of 5 min. A total of 116,927 pieces of vehicle flow data with 5-min intervals in the past 406 days were taken as the data set for this XGBoost time series prediction. In order to enrich the feature vectors, we built some lags on the data set, and the number of lags is the number of feature vectors in this experiment. The number of feature vectors not only affects the degree of fitting but also affects the operation speed of the algorithm. Therefore, we should first process the data appropriately and select the appropriate number of lags as the feature of this prediction.

Usually, in the time series data, typically built of 8–128 lags to machine learning characteristic vector. In Bin Sun's 2018 paper [20], he tried to model different amounts of lags in the XGBoost algorithm, let XGBoost algorithm itself for the importance of these characteristic variables selection, shows the lag takes different values, the impact on the modelling accuracy. We found that when the number of lags was 144, it basically covered all the useful feature vectors of the traffic flow data of the time series for the XGBoost algorithm, and the calculation amount was not large. Therefore, 144 lags (144 eigenvectors) were used for this prediction. Each lag is subtracted 5 min from the previous lag, which constitutes a basic method to predict the traffic flow data at the next time with the characteristics of 144 data (data of half a day in the past). Finally, the feature vectors are given to the XGBoost algorithm for learning, and the prediction is given (Fig. 1).

In this experiment, we configured the way of cross-validation. For each prediction, three times of cross-validation were needed to output the final predicted value. In addition, we compared the influence of different parameters in XGBoost on the accuracy of our prediction in this experiment.

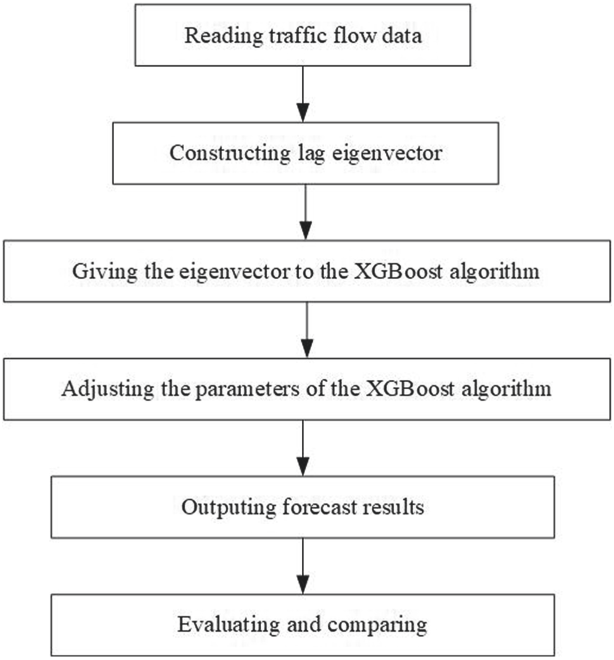
Reading traffic flow data
Constructing lag eigenvector
Giving the eigenvector to the XGBoost algorithm
Adjusting the parameters of the XGBoost algorithm
Outputing forecast results
Evaluating and comparing

Fig. 1. The logical diagram shows the steps of our experiment. Extract the original data, construct the feature vector, and then send the feature vector to XGBoost, and adjust the XGBoost related parameters to output the prediction results, and finally evaluate.

4 Results

The prediction of time series is a typical regression algorithm in machine learning. For the evaluation of the regression algorithm, we chose MSE, RMSE, MAE and R-squared as four indexes. The formula is as follows:

$$MSE = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2$$

$$RMSE = \sqrt{\frac{1}{m}\sum\nolimits_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2}$$

$$MAE = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left|\left(y_i - \hat{y}_i\right)\right|$$

$$R^2 = 1 - \frac{\sum\nolimits_i\left(\hat{y}_i - y_i\right)^2}{\sum\nolimits_i\left(\hat{y}_i - y_i\right)^2}$$

In order to explore the influence of different parameters in XGBoost on the accuracy of traffic flow data prediction. We set several groups of different parameters for the XGBoost algorithm and made predictions under the same data set. The predicted results are as follows (Table 1):

Table 1. Impact of Booster parameters on predicted results

Booster	GBTree	Gbliner
MSE	64.8627	69.7946
MAE	5.7458	6.0019
RMSE	8.0537	8.3543
R-squared	0.9060	0.8988

Booster parameters are used to select the base model for each iteration. GBTree is a tree-based model and Gbliner is a linear model. We found that GBTree model was more accurate in predicting traffic flow data. R-squared is closer to 1 (Table 2).

Table 2. The effect of the Max_depth parameter on the predicted results.

Max_depth	3	5	7	9
MSE	62.7124	63.6834	66.4980	69.6052
MAE	5.6909	5.7103	5.8174	5.9586
RMSE	7.9191	7.9802	8.1546	8.3430
R-squared	0.9090	0.9077	0.9036	0.8991

Max_depth is the maximum depth of the tree. Used to control the overfitting of the model. The higher the value of Max_depth is, the more specific model learning is. From the table, we can see that when Max_depth is 3, the prediction effect is the best. With the increase of the value, the prediction accuracy decreases (Table 3).

Table 3. The effect of the Min_child_weight parameter on the predicted results.

Min_child_weight	1	2	3	5	6
MSE	64.8627	64.2205	65.0793	64.8114	64.7080
MAE	5.7458	5.7456	5.7620	5.7538	5.7567
RMSE	8.0537	8.0138	8.0671	8.0505	8.0441
R-squared	0.9060	0.9069	0.9057	0.9060	0.9062

Min_child_weight is the sum of the minimum sample weights, which is used to avoid overfitting. When its value is large, the model can avoid learning local special samples, but it may also lead to under-fitting. In this experiment, when the value of Min_child_weight is 3, the algorithm performs the best, and the value of R-squared is 0.9069. We find that the high or low value of Min_child_weight will slightly affect the accuracy of prediction (Table 4).

Table 4. The effect of the N_estimators parameter on the predicted results.

N_estimators	10	20	50	100	200	400
MSE	65.4463	62.8453	63.6407	64.8627	66.7749	68.9352
MAE	5.7727	5.6796	5.6986	5.7457	5.8298	5.9387
RMSE	8.0899	7.9275	7.9775	8.0537	8.1716	8.3027
R-squared	0.9051	0.9089	0.9077	0.9060	0.9032	0.9001

N_estimators is the number of the largest tree generated and also the maximum number of iterations. The experimental results show that when the value of N_estimators is 20, the prediction effect is the best, and R-squared reaches 0.9089. When the number of iterations is between 20 and 50, the best effect is achieved. However, when the number of iterations is too large or too small, the prediction accuracy of the XGBoost model for traffic flow data drops slightly.

5 Conclusion

In this experiment, we predict the traffic flow data based on XGBoost algorithm. We take the traffic flow of a fixed road in a 5-min interval in 406 days as the original data set and select the traffic flow data of the past half-day as the feature each time to predict the traffic flow of the next moment. Experimental results show that the XGBoost algorithm can effectively adapt to the traffic flow data, and the prediction effect is good. The value of R-squared is generally around 0.9. After that, we compared the influence of three very important basic parameters in the XGBoost algorithm on the experiment effect. In general, the adjustment of parameters has little influence on the accuracy of the prediction of this data set, with only some small changes. It turns out that XGBoost itself is pretty well optimized. In the future, we will conduct further exploration in two aspects. On the one hand, we will select more traffic flow data sets of different roads to train our model and improve the accuracy and application range of the model. On the other hand, continue to study the influence of different parameters in XGBoost on traffic flow data prediction.

References

1. Jie, L., Van Zuylen, H.J.: Road traffic in China. Procedia Soc. Behav. Sci. **111**, 107–116 (2014)
2. Liu, Z., Yue, X., Zhao, R.: The cause of urban traffic congestion and countermeasures in China. Urban Stud. **11**, 90–96 (2011)
3. Sun, B., Cheng, W., Goswami, P., Bai, G.: An overview of parameter and data strategies for k-nearest neighbours based short-term traffic prediction. In: ACM International Conference Proceeding Series 2017, pp. 68–74. ACM (2017)
4. Wen, H., Sun, J., Zhang, X.: Study on traffic congestion patterns of large city in China taking Beijing as an example. Procedia Soc. Behav. Sci. **138**, 482–491 (2014)

5. Sun, B., Cheng, W., Bai, G., Goswami, P.: Correcting and complementing freeway traffic accident data using Mahalanobis distance based outlier detection. Tehnicki Vjesnik-Technical Gazette **24**(5), 1597–1607 (2017)
6. Sun, B., Ma, L., Shen, T., et al.: A robust data-driven method for muti-seasonal and heteroscedastic IoT time series preprocessing. In: Wireless Communications and Mobile Computing (WCMC), p. 6692390 (2021)
7. Li, J., Walker, J.L., Srinivasan, S., et al.: Modeling private car ownership in China: investigation of urban form impact across megacities. Transp. Res. Rec. **2193**(1), 76–84 (2010)
8. Lv, Y., Duan, Y., Kang, W., et al.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2014)
9. Chen, Y., Shu, L., Wang, L.: Traffic flow prediction with big data: a deep learning based time series model. In: 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1010–1011. IEEE (2017)
10. Ma, L., Sun, B., Han, C.: Learning decision forest from evidential data: the random training set sampling approach. In: 4th International Conference on Systems and Informatics (ICSAI), Hangzhou, China (2017)
11. Ahmed, M.S., Cook, A.R.: Analysis of Freeway Traffic Time-Series Data By Using Box-Jenkins Techniques (1979)
12. Shi, D., Ding, T., Ding, B., et al.: Traffic speed forecasting method based on nonparametric regression. Comput. Sci. **43**(2), 224–229 (2016)
13. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
14. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat., 1189–1232 (2001)
15. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
16. Sagi, O., Rokach, L.: Ensemble learning: a survey. Wiley Interdiscip. Rev. Data Min. Knowl. Disc. **8**(4), e1249 (2018)
17. Lewis, R.J.: An introduction to classification and regression tree (CART) analysis. In: Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, California, p. 14 (2000)
18. Ma, L., Sun, B., Li, Z.: Bagging likelihood-based belief decision trees. In: 20th International Conference on Information Fusion (FUSION), Xi-An, China, pp. 1–6 (2017). http://ieeexplore.ieee.org/abstract/document/8009664/
19. Bickel, P.J., Li, B., Tsybakov, A.B., et al.: Regularization in statistics. TEST **15**(2), 271–344 (2006)
20. Geng, R., Sun, B., Ma, L., Zhao, Q., Shen, T.: Anomaly-aware in sequence data based on MSM-H with EXPoSE. In: 40th Chinese Control Conference (CCC 2021), Shanghai, China (2021)
21. Sun, B., Cheng, W., Goswami, P., et al.: Short-term traffic forecasting using self-adjusting k-nearest neighbours. IET Intel. Transp. Syst. **12**(1), 41–48 (2018)