Time series forecast of sales volume based on XGBoost

Lingyu Zhang1,2, Wenjie Bian2, Wenyi Qu2, Liheng Tuo2,3 and Yunhai Wang1*
1 School of Computer Science and Technology, Shandong University, Qingdao, China
2 Didi AI Labs, Didi Chuxing, Beijing, China
3 Smart Transportation Center, National Engineering Laboratory for Big Data Analysis and Applications, Beijing, China
805906920@qq.com, bianwenjie_i@didiglobal.com, quwenyi_i@didiglobal.com, tuoliheng@didiglobal.com, cloudseawang@gmail.com

Abstract. Some problems such as the decline of new labor force, the increase of retired labor force emerge because of the complex and changeable market environment, consequently exacerbating the staffing problem in the retail industry. Also, the unreasonable distribution of personnel, there are few people in busy hours and too many people in idle hours, which causes waste of labor. To address this issue, we analyzed the time series of sales volume in the retail industry in detail, and processed the data with feature engineering for predicting the in-store sales volume in the future. At the same time, other features such as weather and temperature are added to improve the accuracy of the model. Considering the characteristics of the data, we choose XGBoost as the prediction model. The experiments on real-world datasets verified better performance of proposed model compared with other state-of-the-art models.

1. Introduction
Time series forecasting of sales volume is an important application in time series forecasting. In the age of big data, it is easier than ever to obtain detailed information about each transaction. If the sales volume in the next period of time can be predicted based on historical transaction information, the managers of the retail store can give a more reasonable staffing schedule, avoiding misarranging too many people during slack periods or too few people during peak periods.

As to the problem of time series forecasting, linear models and ARIMA [6] are widely used. With the in-depth research on machine learning and deep learning, new directions are available for time series forecasting. The SVM algorithm has been known as a classic algorithm for the prediction of financial time [10, 1, 7]. Jain et al.[5] combined the traditional time series method and neural network to predict the time series, showing advantages of both methods. Zhang et al. proved the effectiveness of neural networks for linear time series forecasting[15], and used neural networks to model and analyze seasonal time series with trend components[14], and then proposed a hybrid model of ARIMA and neural network to predict time series[13]. Liu[12], Siami-Namini[9], Sagheer[8] et al. verified great performance in predicting time series using LSTM algorithm. In this paper, in addition to applying feature engineering to filter the time series characteristics of the data set itself, we additionally introduced objective environmental factors such as weather and temperature to improve the accuracy of sales volume predicting. By analyzing the features of data set in detail, XGBoost[2] is chosen as the training model to predict the time series of sales volume and would be compared with other state-of-the-art models in experiments.
The rest of the paper is organized as follows. We describe the time series of sales volume datasets and performs feature engineering in Section 2. Section 3 introduces the principle of XGBoost. Evaluations on two different datasets are presented in Section 4. Section 5 contains the conclusion.

2. Data description

This section introduces the raw data used for sales volume forecasting. The raw data includes two datasets of trading orders of two entity stores of the same milk-tea brand and the records of weather in Beijing, which is of year 2019. Table 1 shows the statistical information of the two original datasets and the weather_data. To be more specific, data1 contains 1,187,609 orders within a full year from January 19th of 2019 to January 19th of 2020. Each order contains company number, store number, order number, billing time, business settlement time, turnover, payment method and cash register number. And data2 contains 89109 transaction in seven months from July 17, 2019 to January 19, 2020. Billing time, business settlement time and order confirming time are all the same. Turnover is the actual transaction amount of this order. The weather_data contains a total of 396 weather records covering the weather condition of every day from January 1, 2019 to January 31, 2020, including date, maximum temperature, minimum temperature, weather conditions, wind direction, air quality, etc.

The orders in the original data are recorded based on time, but our model is to predict the sales volume in a certain period of time, so we have to process the original data. We divide the time period according to the granularity of fifteen minutes, add the order value in each time period as the sales volume and the total number of orders as the order volume. In this way, we have divided three feature, timestamp, sales volume(gmv), and order volume. Analyzing the quantiles of the two new datasets, We found that the data above 95% quantiles are very scattered, accompanied by some data with actual value below 0. We treat these data as abnormal values and perform truncation processing on these abnormal values, making the model pay more attention to conventional data and preventing problems such as large estimates caused by abnormal values. Then we check distribution analysis of the processed datasets.

2.1. Distribution of trading orders

Figure 1. (a)-Figure 1. (d) and Figure 2. (a)-Figure 2. (d) reveal the distribution of the two datasets respectively. The sales volume of the two entity stores is different for the various regional consumption level. However, the orders of sales contribution of two stores look the same. And considering the similarity in contribution, we mainly analyzed the data1 as example. Figure 1.(a) is the distribution of sales volume and order volume at one-week intervals. From the figure, we can conclude that the sales volume and order volume are both distributed evenly during the workdays, but the significantly surge on weekends, and reach the peak on Saturday. Figure 1.(b) and Figure 1.(c) show the average daily volume distribution and the average monthly volume distribution. From the figure, we can conclude that there are some ups and downs of the sales volume of each month. Though, there are two peaks in monthly distribution. First, there will be a lower local maximum between April and May, and then it will reach the peak between October to December. The lowest point of the volume is near the beginning of January. Figure 2.(d) describes the sales volume distribution on holidays, workdays and daily with a granularity of 15 minutes. From the figure, we can conclude that the sales volume of holidays and weekends are significantly higher than that of workdays, and the fluctuating trends of weekends/holidays, workdays and all days seem similar. The sales volume reaches its maximum at ten o'clock, when the business just started, then the trend gets smoothing from at 10:30 to 19:00, and then declines linearly after 19:00. Order_data2 sharing the distribution and changing trends above, but for different regional consumption level in detail, the peak sales volume of order_data2 occurs in July and December.

| Table 1. The information of data. |
|-------------------------------|----------------|---------------|------------|
| Original data                | Start          | End           | Size       |
| order_data1                  | 2019.1.19      | 2020.1.19     | 1187609    |
2.2. Weather data

We analyzed the correlation between sales volume and some features of weather like temperature, weather conditions (Rainy or not), wind direction, air quality and etc. and found that air quality has a strong correlation with sales volume. Other features such as temperature, weather conditions, wind direction have relatively much more weak correlation, and we do not specifically list the details of the analysis here. When analyzing air quality, according to the air quality index, we divided air quality into three levels (excellent, average, and poor) and analyzed the three levels separately.

Figure 3.(a)-Figure 3.(c) respectively describe the distribution of air quality in Beijing for the whole year of 2019 and January of 2020 and the distribution of sales volume of two datasets under different air quality conditions. From Figure 3.(a), we can see that the air quality in Beijing is relatively poor in spring, among the air quality of March is the worst. Autumn and winter are the next second, and the air quality in summer is the best. When the air quality is in level poor, the sales volume is better than that in other two levels as excellent and average. And there is no obvious difference between the sales volume with air quality in level excellent and average (Figure 3.(b), Figure 3.(c)).
Figure 2. Distribution of order\_data2 (a) Distribution of gmv and orders within a week; (b) daily mean distribution within one year; (c) Monthly mean distribution within one year; (d) holiday & workday

Figure 3. (a) Distribution of air quality in Beijing for the whole year of 2019 and January 2020; (b)-(c) Distribution of sales volume of two datasets under different air quality conditions
2.3. Feature engineering

According to the data analysis above, we established a time series with the unit of day, month, and year, day of the week, hour, minute, holiday, N-day historical transactions sequence, and selected the air quality as key feature. From Figure 2.(a) and Figure 3.(a), we got the best statistical characteristics from the 7-day period cycle, so the value of N is selected as 7. Due to the strong correlation between sales volume feature and order volume feature, we will not reduce redundant features referring to the order volume. For feature description in detail please refer to Table 2.

| Feature                          | Description                                           | Scope |
|----------------------------------|-------------------------------------------------------|-------|
| Month                            | The month of the forecast period                      | 1-12  |
| Day of the month                 | Which day is the forecast period of the month         | 1-31  |
| Day of the week                  | Which day is the forecast period of the week          | 0-6   |
| Hour                             | Hour of forecast period                               | 0-23  |
| Minute                           | Minute of forecast period                             | 0-45  |
| Holiday                          | Whether the forecast period is a holiday              | 0,1   |
| 7-day historical transaction sequence | The 7-day transaction sequence closest to the forecast period | -     |
| Air quality                      | Air quality at the forecast period                   | 0,1,2 |
| Highest temperature              | Highest temperature at the forecast period           | -     |
| Lowest temperature               | Lowest temperature at the forecast period            | -     |
| Rainy                            | Whether there is rain during the forecast period     | 0,1   |
| Hazy                             | Whether there is haze during the forecast period     | 0,1   |

3. Methodology

The XGBoost algorithm is a gradient boosting framework created by Chen et al.[2] in 2016. It is widely used in various data mining scenarios and algorithm competitions. The algorithm takes its main advantages in precision, flexibility, and automatic processing of missing values.

3.1. Gradient Boosting

Gradient Boosting is a machine learning method as a linear additive model composed of an ensemble of weak prediction models. It take of $M$ steps to obtain the complete model $F$. The model $F_m$ will not be optimized directly at step $m + 1$. Otherwise the basic model $h_{m+1}(x)$ will be trained and to calculate the residual value $y - F_m$ for the prediction model at step $m + 1$, approaching the value $\mathcal{Y}$.

$$F_{m+1} = F_m + h_{m+1}(x)$$

(1)

Therefore, calculating aim turns to be how to find $h_{m+1}(x) = F_{m+1} - F_m$. Generally, the negative gradient of the objective function is used as the residual to learn the basic model $h(x)$.

3.2. XGBoost

XGBoost is an implementation of Gradient Boosting, which combines multiple weak classifiers into a strong classifier in a linear way. XGBoost supports both CART and linear classifiers as base classifiers, and performs second-order Taylor expansion on the cost function, expressing more plentiful information. In terms of operating speed, XGBoost supports parallel selection of split points, and the model training cost much less time.
The main idea of XGBoost is to continuously append weak trees with different weights to the set. The trees in the set have to approach the residuals of the previous prediction as much as possible, which is expressed as follows

\[ \hat{y}_i = \sum_{k=1}^{K} f_k(x_i) \quad f_k \in F \]  

(2)

In (2), \( \hat{y} \) is the predicted value, \( F \) is the set including all regression trees, \( f_k \) is one of the regression trees, and \( K \) is the number of regression trees. The predicted value \( \hat{y}_i \) is expected to be as close to the true value \( y_i \) as possible, and meanwhile without losing its generalization ability. The formula to compute Obj is below

\[ \text{Obj}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^{T} \Omega(f_t) + \text{constant} \]  

(3)

In (3), \( l(y_i, \hat{y}_i^{(t)}) \) is the loss function, which represents the difference between the predicted value and the true value. It can be any form of loss function which is second-order derivable. \( \Omega(f_t) \) is the regularization term, which defines the complexity of the model. The smaller the value of \( \Omega(f_t) \), the lower the complexity and the stronger the generalization ability.

\[ \Omega(f) = \gamma T + \frac{1}{2\lambda}||w||^2 \]  

(4)

In (4), \( T \) is the number of leaf nodes, and \( \mathcal{W} \) is the score represented by the leaf nodes. XGBoost uses second-order Taylor expansion to expand the loss function in the gradient boosting process. The final objective function is as follows

\[ \text{Obj}^{(t)} \approx \sum_{i=1}^{n} \left[ g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \]

\[ = \sum_{i=1}^{n} \left[ g_i w_q(x_i) + \frac{1}{2} h_i w_q^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} w_j^2 \]

\[ = \sum_{j=1}^{T} \left[ (\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T \]

\[ g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \quad h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \]  

(5)

Where \( g_i \) and \( h_i \) are the first-order derivative and second-order derivative of each data point in the error function, and \( I_j \) is the index set of the samples on each leaf node \( j \)

\[ I_j = \{ i | q(x_i) = j \} \]  

(6)

For given \( q(x_i) \), taking the derivative of \( w_j \) equal to 0 can get the best weight \( w_j^* \) of leaf node \( j \)

\[ w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \]  

(7)

Calculate the optimal value by the following formula

\[ L^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} - \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} - \gamma \]  

(8)
4. Experimental study
This section mainly compare XGBoost with other state-of-the-art models on the chosen datasets.

4.1. The experimental setup
We evaluated the performance of the model on order_data1 and order_data2. In chronological order, we use the first 75% of the dataset as the training set to train the models, and the last 25% of the dataset as the test set to test the performance of the models.

4.1.1. Baselines. We compare XGBoost with several state-of-the-art models. The introductions to these models are below:
- ARIMA. The ARIMA algorithm is a classic time series forecasting algorithm. We have checked that the dataset is a stationary series, simply applied our dataset to ARIMA without preprocessing.
- LSTM. The LSTM[4] algorithm was first proposed by Hochreiter and Schmidhuber in 1997. It is a specific form of RNN model. LSTM is the most widely used time series prediction algorithm in deep learning.
- Prophet. The Prophet[11] algorithm is an open sourced time series forecasting algorithm developed by Facebook in 2017. It not only can automatically process some abnormal values and missing values on the time series, but also can automatically predict the future trend of the time series.
- GBDT. The GBDT[3] algorithm is an iterative decision tree algorithm, including regression algorithms and classification algorithms, belonging to the Boosting algorithm family. It can also automatically handle missing values and abnormal values.

4.1.2. Metrics. We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the performance of the model.

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - y'_i|
\]

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

\[N \] is the number of samples, \( y_i \) is the true value of the \( i \)th sample, and \( y'_i \) is the predicted value of the \( i \)th sample.

4.2. Experimental result
Table 3 shows performance of proposed model and other models on two datasets of the entity stores. Among them, the GBDT model of order_data1 was iterated 900 times, and the XGBoost model was iterated 320 times; the GBDT model of order_data2 was iterated 400 times, the XGBoost model was iterated 130 times. From the table, we can get the following conclusions: (1) Among the five models, the GBDT and XGBoost achieve the best performance, while the ARIMA method has the worst performance. (2) Although the results of GBDT and XGBoost are similar, XGBoost only needs 1/3 of the iterations of GBDT to achieve a better prediction performance than GBDT.

| datasets   | Method | RMSE  | MAE   |
|------------|--------|-------|-------|
| order_data1| ARIMA  | 0.285868 | 0.226055 |
|            | LSTM   | 0.243492 | 0.188715 |
|            | Prophet| 0.247893 | 0.206457 |
4.3. Result analysis
The following analysis explains why the XGBoost method performs better than other methods in the time series forecast of transaction problem. (1) Compared with deep learning methods, boosting methods (GBDT, XGBoost) demand for fewer data and fewer features. We aim to using the information from the previous period to predict the sales volume in the next period of time. The data and features we have access small, so the GBDT and XGBoost perform better than the other models. (2) Compared with GBDT, the XGBoost explicitly adds a regularization term to the objective function to control the complexity of the model, prevent overfitting, and improve the generalization ability of the model. In addition, XGBoost performs Taylor's second-order expansion of the objective function during training. It can use both the first-order derivative and the second-order derivative information, which significantly improved the performance over GBDT.

5. Conclusion
Predicting the sales volume with desired accuracy in the next period of time based on the historical sales volume sequence can save cost for staffing. In this paper, we specifically analyzed the feature of the sales volume sequence, and added external feature of weather data such as air quality, highest temperature, lowest temperature to improve accuracy. Based on the feature mentioned above, we selected XGBoost as the prediction model and achieved excellent performance.

Acknowledgments
We are grateful to anonymous reviewers for their helpful comments. This work is supported in part by NSFC Grant No. 61772315 and 61861136012, and National Key R&D Program of China (No.2018AAA0101100).

References
[1] Cao, L.J., Tay, F.E.H.: Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on neural networks 14(6), 15061518 (2003)
[2] 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)
[3] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
[4] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
[5] Jain, A., Kumar, A.M.: Hybrid neural network models for hydrologic time series forecasting. Applied Soft Computing 7(2), 585–592 (2007)
[6] Jakaša, T., Andročec, I., Sprčić, P.: Electricity price forecasting—arima model approach. In: 2011 8th International Conference on the European Energy Market (EEM). pp. 222–225. IEEE (2011)
[7] Kim, K.j.: Financial time series forecasting using support vector machines. Neurocomputing 55(1-2), 307–319 (2003)
[8] Sagheer, A., Kotb, M.: Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing 323, 203–213 (2019)

[9] Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of arima and lstm in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). pp. 1394–1401. IEEE (2018)

[10] Tay, F.E., Cao, L.: Application of support vector machines in financial time series forecasting. omega 29(4), 309–317 (2001)

[11] Taylor, S.J., Letham, B.: Forecasting at scale. The American Statistician 72(1), 37–45 (2018)

[12] Yunpeng, L., Di, H., Junpeng, B., Yong, Q.: Multi-step ahead time series forecasting for different data patterns based on lstm recurrent neural network. In: 2017 14th web information systems and applications conference (WISA). pp. 305–310. IEEE (2017)

[13] Zhang, G.P.: Time series forecasting using a hybrid arima and neural network model. Neurocomputing 50, 159–175 (2003)

[14] Zhang, G.P., Qi, M.: Neural network forecasting for seasonal and trend time series. European journal of operational research 160(2), 501–514 (2005)

[15] Zhang, G.P.: An investigation of neural networks for linear time-series forecasting. Computers & Operations Research 28(12), 1183–1202 (2001)