A Study on Prediction Algorithms for Yu'EBao Capital Flow

Xie Chunyang, Pan Zicen, Zhang Yuqi

Abstract

With the development of Internet Finance, money market funds such as Yu'EBao have become new fashion of small-scale investments. These funds meet large demands of purchase and redemption everyday, and see liquidity and position management as a vital issue. In order to manage liquidity effectively and use the rest of the capital in investments, fund managers need to predict the capital inflow and outflow in the near future. In order to find the best solution for practical use, in this paper we take desensitized full behavior log of 30,000 Yu'EBao users over the course of 13 months, apply regression and time series analysis to the data, ensemble the models, and use our own design of features to predict the total amount of purchase and redemption per day in the following month. Then we compare our predictions with actual values, and give feasibility reports of the models in this problem. Among the algorithms we tested, the ensemble of Neural Network and GBDT regression worked the best.

Key Words: Big Data, Machine Learning, Yu'EBao, Regression, Time Series, Neural Network

Table of Contents

1. Introduction

In the recent years, Internet Finance gradually becomes one of the most important parts of finance system. The funds which people can deposit or withdraw at any time inevitably encounter large amounts of capital inflow or outflow everyday. Yu'E Bao, the most paramount one of this kind of funds, is an investment product offered through the Chinese e-commerce giant Alibaba Group Holding Ltd.'s third-party payment affiliate, Alipay.com Co. With such a huge user base, the liquidity management of capital becomes a thorny problem to the company. Alibaba, as a profit-oriented organization, needs not only to minimize the liquidity risks of funds, but also to keep a net earning of the investment thus to maximize profits. Therefore, accurate prediction of the capital inflow and outflow is extremely important to the company. (For Monetary Funds, capital inflow means purchase of funds by clients, and capital outflow means redemption of funds.) Accurate predictions of the daily capital flow of funds would enable the company to adjust its investment strategies to achieve both safety and profit maximization.

The work of this paper comes from the participation of Chunyang Xie, one of the authors, in Yu'EBao Capital Inflow and Outflow Prediction Competition, one in Alidata Big Data Series Competition organized by Alibaba. This paper analyzes multiple mathematical models applicable to the intended problem, compares different methods and obtains the optimal solution.

2. Problem Overview

2.1 Data Preparation

Ant Financial Co. (operator of Yu'E Bao) provides the complete data of the purchase and redemption of 30,000 users within 13 months (July 2013-Augest 2014) and users' basic personal information. The data have been desensitized and encrypted. This paper is grounded on these data to predict the sum of the purchase and redemption of funds about September 2014 on a daily basis with a precision to cent.

The source data is shown as follows, divided into four tables: User Basic Information Table, User Purchase and Redemption Data Table, Yu'EBao Yield Table and Interbank Offered Rates Table.

Table Name	Column Name	Explanation
User Basic Information Table	user_id	User ID
	Sex	User Gender
	City	User's City
	constellation	User's Constellation
User Purchase and Redemption Data Table	user_id	User ID
	report_date	Date of Transaction
	tBalance	Today's Balance
	yBalance	Yesterday's Balance
	total_purchase_amt	Today's Total Purchase Amount (=Direct Purchase + Yield)
	direct_purchase_amt	Today's Direct Purchase Amount
	purchase_bal_amt	Purchase Amount from Alipay
	purchase_bank_amt	Purchase Amount from Bank
	total_redeem_amt	Today's Total Redemption Amount (=Withdrawal + Consumption)
	consume_amt	Today's Consumption Amount
	transfer_amt	Today's Withdrawal Amount
	tftobal_amt	Withdrawal to Alipay Amount
	tftocard_amt	Withdrawal to Bank Amount
	share_amt	Today's Share/Yield
	category1	Consumption in Category 1
	category2	Consumption in Category 2
	category3	Consumption in Category 3
	category4	Consumption in Category 4
Yield Table	mfd_date	Date
	mfd_daily_yield	Daily Yield Per 10,000 Yuan
	mfd_7daily_yield	Seven-Day Annualized Yield
Shanghai Interbank Offered Rate Table	mfd_date	Date
	Interest_O_N	Overnight Rate
	Interest_1_W	One Week Rate
	Interest_2_W	Two Weeks Rate
	Interest_1_M	One Month Rate
	Interest_3_M	Three Months Rate
	Interest_6_M	Six Months Rate
	Interest_9_M	Nine Months Rate
	Interest_1_Y	One Year Rate

Table 1 Source Data

User Basic Information Table provides the basic information about users. The dataset consists of 30,000 users randomly chosen, including their user ID, sex, city and constellation.
User Purchase and Redemption Data Table provides information on the purchase and redemption of funds by users with the category of transaction over a span of 13 months (July 2013-August 2014). The data have been desensitized and encrypted while maintaining the original trend. The data mainly consists of users' date and details of each transaction. Each entry of users' operations includes both purchase and redemption on a certain date, with the category of transaction.
Yu'EBao Yield Table includes the yield of Yu'EBao (both million-fund return and seven-day annualized rate) during 13 months (July 2013-Augest 2014).
Shanghai Interbank Offered Rates Table provides the lending rates between banks, which have been annualized.

2.2 Evaluation Method

Our results are evaluated by daily total purchase and redemption values in the following 30 days (1st September 2014-30th September 2014). There are 60 data points in total.
Considering actual business scenarios, Alibaba combines the relative error of each single data point with an undisclosed function to evaluate the predication results as follows:
First calculate the relative error between predicted and actual daily purchase and redemption.
The relative error of the purchase on the i^{th}day (real value p_i, predicted value $\widehat{p_i}$)

$$e_i^p = \frac{|p_i - \widehat{p_i}|}{p_i}$$

The relative error of the redemption on the i^{th}day (real valueq_i, predicted value $\widehat{q_i}$)

$$e_i^q = \frac{|q_i - \widehat{q_i}|}{q_i}$$

And the formula of evaluation indicator W is defined as

$$W = \sum_{i=1}^{30} [f(e_i^p) * 0.45 + f(e_i^q) * 0.55]$$

$f(x)$, although undisclosed function, is strictly decreasing, thus the smaller the error, the

higher the score, and vice versa. Because the prediction of redemption is relatively more important than purchase in the actual business scenarios, Alibaba put different weights to purchase and redemption in the formula of evaluation.

Due to the limitations by Alibaba, we can only submit results for evaluation once per day and we cannot correlate accuracy with scores. In addition to this evaluation method, the paper uses August dataset as a test set to evaluate certain models through the relative error of single data points. In the ARIMA model, parameters are selected based on the results of this evaluation method.

The formula of W' is defined as

$$W' = \sum_{I=1}^{30} \left(\frac{|p_i - \hat{p}_i|}{p_i} - \frac{|q_i - \hat{q}_i|}{q_i} \right)$$

The essence of this problem is Big Data analysis and prediction in the field of finance. In the paper, this problem is approached in two ways: Time Series Analysis and Regression. Because the given data and the data to predict are on the same time series, and the relationship between neighboring data points is clear, the use of Time Series Analysis, a common method used in finance, is appropriate. At the same time, every date has its own unique features. Assuming that relationship between features and capital flow exist, we use multivariate regression to offer an alternative solution. Detailed analysis of model selection and comparison between different mathematical models are elaborated in Section 4 of this paper.

3. Data Observation

One important idea in Big Data Analysis is that the appropriate method of solution is determined by features of the data rather than by human hypothesis.

This paper first introduces the distribution of data, categorization of users, user behavior patterns and relevant features based on four data tables above. Based on the assumption that these features are correlated with actual values to predict, this paper presents five prediction methods and feature selection criteria. (See Section 4)

3.1 Patterns in User Purchase and Redemption Data Table

Large differences exist in transaction frequency and quantity among different users. Most of the 30,000 users have few transactions. 84.4% of the users have less than 10 entries of transactions, and as many as 45.2% of users never purchased or redeemed (only registered an account). The users with total transaction records of more than 50,000 yuan only accounts for 5% of users, while constituting 70% of the daily transaction amount on average. It is hard to predict the next transaction time and amount for users with total transaction less than 1,000 yuan. The use of randomization would be reasonable but further lowering accuracy. For users with over 50,000 yuan total transactions, it is common to see sudden purchase or redemption by a large amount, making it hard to predict. Therefore, it is not feasible to predict and congregate transactions by user.

Undoubtedly, users with few transaction records can be disregarded conditionally, and predictions on users with frequent transactions should be the priority.

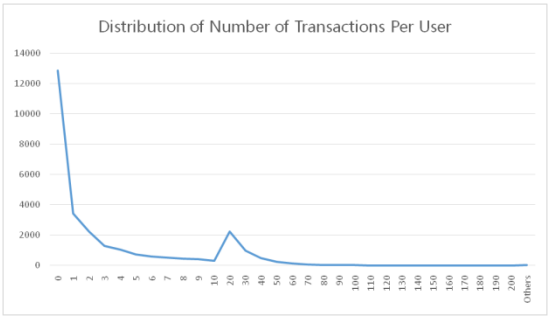


Fig.1(a) Distribution of Number of Transactions Per User

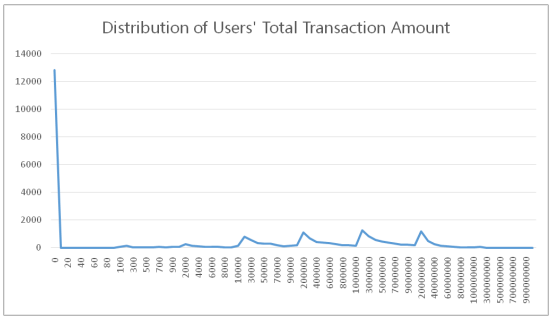


Fig.1 (b) Distribution of Users' Total Transaction Amounts

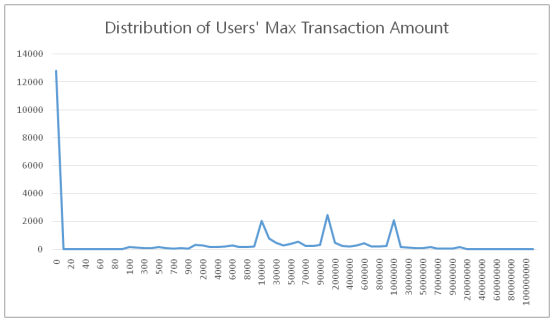


Fig.1(c) Distribution of Users' Max Trans. Amt

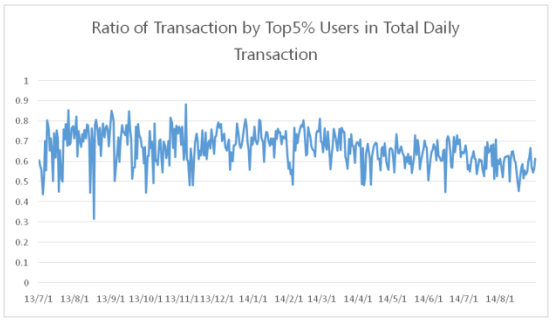


Fig.1(d) Ratio of Trans by Top 5% users in Total Daily Trans.

3.2 Patterns in Total Purchase and Redemption by Day Table

In order to predict the daily total purchase and redemption in the following month, this paper draws a graph of month average of daily total purchase and redemption values.

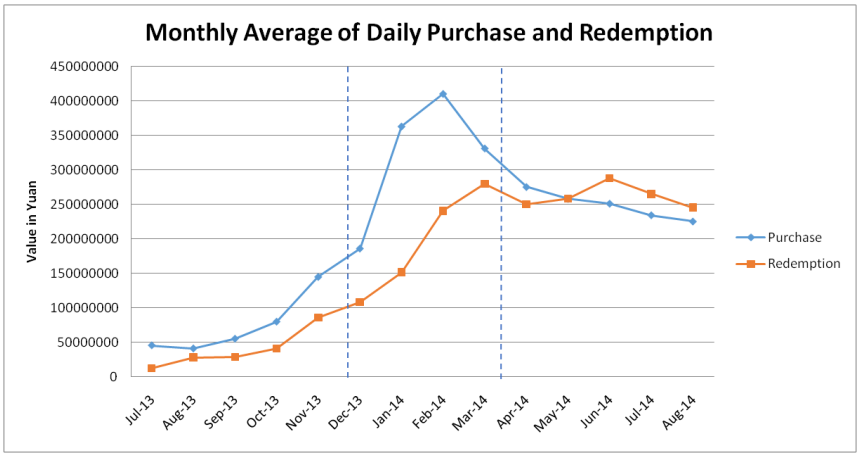


Fig.2 Monthly Average of Daily Purchase and Redemption

From this table, a clear trend in the data is visible.

3.2.1 General Trend

Daily transaction data in this problem can be roughly split into three phases.

The First Phase: 1st September 2013- 31st December 2013. Purchase and redemption values are relatively low but clearly increasing. This can be explained by that Yu'EBao fund had just been established and was in the stage of user accumulation. Data in this phase has no obvious pattern and significant randomness coming from the scarcity of users.

The Second Phase: 1st Jan 2014- 28th Feb 2014. Purchase and redemption values reach their peaks in this period and decrease thereafter. The peak values are likely caused by the great shopping and "red packet money" demands during the Spring Holiday in China.

The Third Phase: 1st Mar 2014- 31st Aug 2014. Purchase and redemption values are generally lower than the second phase. Saturation of user numbers give rise to a relatively stable pattern over time.

Considering the need for stationarity in time series analysis and regression, this paper uses the data in the third phase (1st Mar 2014- 31st Aug 2014) as input to the models and uses other data as a reference to users' behavior patterns.

3.2.2 Seasonal Changes

Observation of the relatively stationary third phase finds that, both daily purchase and redemption values fluctuate in a period of seven days. Such a periodicity matches the common sense – people may have different transaction behaviors on different days in a

week. As shown in Fig.3, the average difference among different days in a week is considerable.

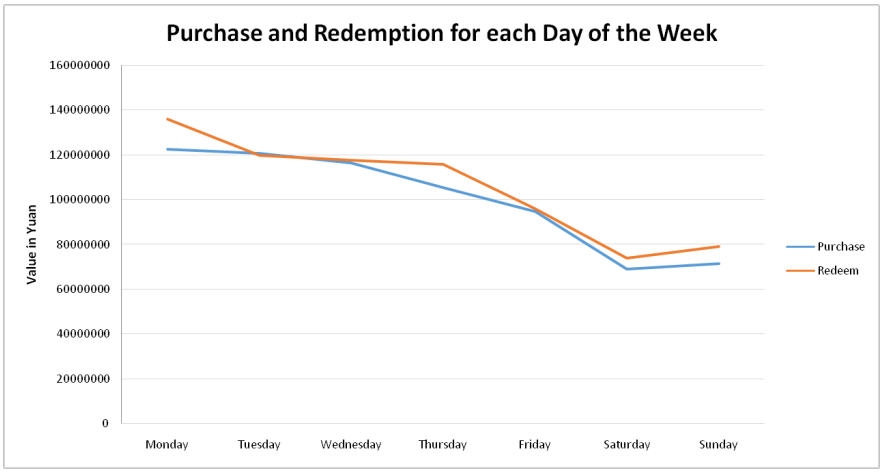


Fig.3 Purchase and Redemption for each Day of the Week

Such a periodicity is also visible in the first and second phase, and this consistency demonstrates the importance of this feature.

Our scope was expanded to consider difference among days in a month, but no clear correlations are found. Thus, this paper added month as a feature in a special way, which will be elaborated in detail in Section 5.

3.2.3 Special Dates

It is easy to understand that people's transaction behaviors during holidays are different from that in workdays. According to the Fig.4, daily purchase and redemption during three-day holidays are lower than that on the same day in the week before and after holiday. The dotted lines on the graph represent the average of daily purchase and redemption values in seven days before and after the holiday.

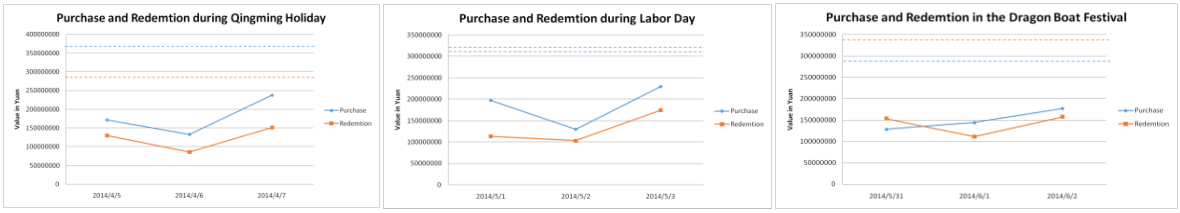


Fig.4 Purchase and Redemption during Holidays

Besides national holidays, dates like Double Eleven when shopping and "red packet" transfer are common, clear deviations can be seen on the data. For example, on Aug 26, 2014, promotions and IPOs might have caused the capital flow to be greater than usual, but

due to the lack of information on the Internet, the actual cause of the abnormality is hard to conclude, and the models in this paper can only not make any changes with respect to these special dates.

4. Problem Analysis

4.1 Two Approaches

In this paper, the problem of capital flow prediction is approached in two ways: Time Series Analysis and Regression. The key to the prediction is to discover the relations among the data points and the patterns within the data. Time Series Analysis is a commonly used method in economics and finance, the basic idea of which is to predict a future data point with the calculation of a few preceding values. The precondition of prediction is that all the data points should not only be in the same time series, but also have contextual influence and relations with each other. Following the conclusions reached in Section Three, this paper transforms the data table into a daily total purchase and redemption table, meeting the requirements of time series analysis. This approach grasps well the trend and fluctuation of data.

Regression analysis demands human observation and understanding of data to summarize the patterns hidden in the data, and base on which find the factors, also known as features, underlying the change of the variables to predict. Then by adding the features into regression formulas or the decision trees, iterations would help determine the optimal parameter combinations to reflect the relationship between features and dependent variables, and to obtain prediction results. Regression analysis is one of the most widely used methods in machine learning.

4.2 Model Selection

The most widely used algorithm in Time Series Analysis is autoregression, which is also in essence regression. In autoregression, values preceding the data point to predict are added as features into regression models for prediction. In practical uses, autoregression often comes with differential treatment and moving average(MA) algorithms. Together they form ARIMA model (autoregressive integrated moving average model), and this combination helps the originally linear equation to adapt to more datasets.

This work selects three basic regression models: Linear Regression, Neural Network Model and GBDT Regression, and uses each of them for autoregression and regression with manually selected features to make predictions. After testing these three models and comparing results, the paper experiments model ensemble to further improve the accuracy of prediction. At last, best result is achieved from ensemble.

5. Feature Selection in Regression Models

Although the three models chosen in this paper have different theories and characteristics, the feature addition methods in all three models are identical. In this paper, multiple combinations of features addition have been tested for each model in practice. The following rules have been observed:

(a) Only add time-related features

For every feature used in regression models, a value needs to be known for each data point to predict in September. However, for features related to finance and user behavior, no actual record is available for use, so the only way to add these features is to predict them with autoregression first. The composition of prediction hinders the accuracy of prediction. Therefore, this paper discarded finance-related features after experiments of using interest rate and yield rate.

Finally, according to observation and understanding of the data, every feature added in this paper is related to time, so that for September dates on which we predict, values of the features are certain and accurate. The features selected in this paper include: Day of the Week, Month Trichotomy (1/3, 2/3, 3/3), Day of the Month, Holiday or Not, Day in Holiday, Month Start and Month End.

(b) Use "binary flag" addition method

Because all the features added are related to date, which means that there is no numeric relationship, so it would be most appropriate to add those features as "binary flag". If a date satisfies the criterion of a feature, the value would be 1, otherwise it would be 0. For this addition method, one feature may be represented as multiple variables in the model formula.

(c) Add features with strong correlations with the dependent variables and reduce the correlation between features

However, more features do not guarantee better results. On one hand, features may interfere with each other, making it more difficult for the model to judge the relationship between variables in model and cause the prediction accuracy to decrease. On the other hand, computation complexity may increase exponentially to cause curse of dimensionality. In conclusion, we should select and only select the most important features.

In order to achieve this, this work used Pearson Product-Moment Correlation Coefficient to measure the correlation between every independent variable and dependent variable. The formula as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

where $cov(X,Y)$ is the covariance between X and Y.

The result of calculation is as follows:

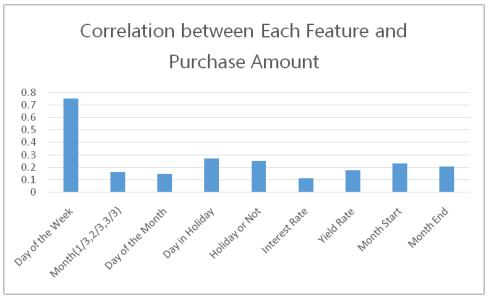


Fig.5(a) Correlation between Each Feature and Purchase Amount

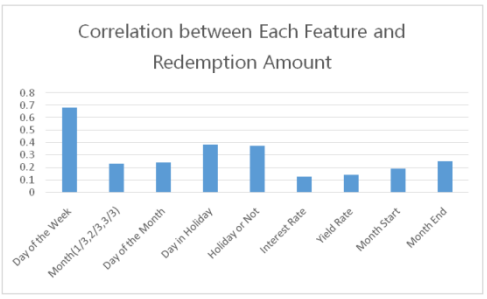


Fig.5(b) Correlation between Each Feature and Redemption Amount

In practice, we referenced Correlation Coefficient, experimented with various feature sets, and selected the optimal configuration according to the evaluation results.

(d) Do not add manually selected features in auto regression model

In the regression models in which we added manually selected features, it is possible to further add previous values of that variable to make it autoregressive. However, this practice may cause complication and repetition in the features, making the prediction disorderly, undermining the prediction accuracy. Therefore, it is not tested in every model. A result of GBDT models with both autoregressive terms and manually selected features is as follows:

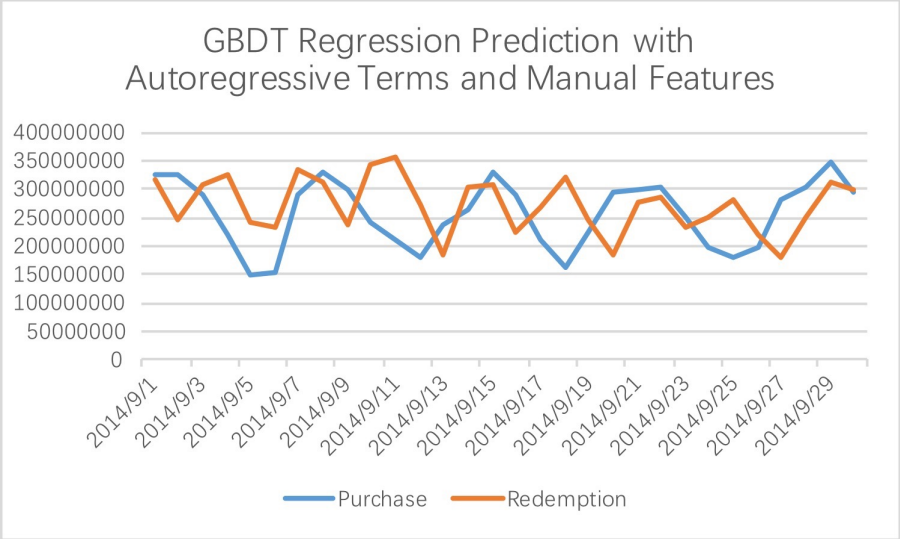


Fig.6 GBDT Regression Prediction with Autoregressive Terms and Manual Features

According to the above observation and conclusions, three models are adopted for more comprehensive tests: linear models, neural network models, and GBDT regression models.

6. Linear Models

6.1 Linear Autoregression (ARIMA) Model

6.1.1 Model Introduction

When linear model is used for antoregression, the model can also be called “Autoregressive Model” (AR). In the common usage of AR models, Moving Average (MA) model and Differential Treatment are often considered together. The ARIMA model (autoregressive integrated moving average model) they form helps the originally linear equation to adapt to more datasets.

Firstly, nonstationary time series is transferred to stationary time series. Then, dependent variable is used to regress its lagged value and the present and lagged value of random error to establish ARIMA model. The central idea is to treat the data series of prediction object based on time series as a random series, and describe and fit this series with a certain mathematic model. By establishing this model, the future values can be predicted with the past values of this series.

ARIMA model is essentially the combination of differential treatment and ARMA model which is established on basis of auto-regression (AR) model and mobile average (MA) model.

AR model can be expressed as follows:

$$X_t = \phi_0 + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t$$

where$\phi_i, i = 1,2,\ldots p$ is auto-regression coefficient, p the order of auto-regression coefficient, and $\varepsilon_t$the white noise sequence. X_t is the auto-regression in pth order which can be represented by $AR(p)$.

MA model can be represented as follows:

$$X_t = \mu - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$$

where $\theta_i, i = 1,2,\ldots q$ is the moving average coefficient, q the order of moving average model and ε_t white noise sequence. X_t is the q-order moving average sequence which can be represented by $MA(q)$.

ARMA model can be expressed as follows:

$$X_t = \mu + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

where $\phi_i, i = 1,2,\ldots p$ is the auto-regression coefficient, $\theta_i, i = 1,2,\ldots q$ the mobile average coefficient and X_t the auto-regression moving average sequence which can be represented by $ARMA(p, q)$.

6.1.2 Stationarization Process

An important precondition to establish ARIMA model is that the series to be predicted should be stationary. In other words, the individual value should fluctuate around the average value of series without significant rising and falling trend. The chart for monthly average of present purchase redemption is as follows.

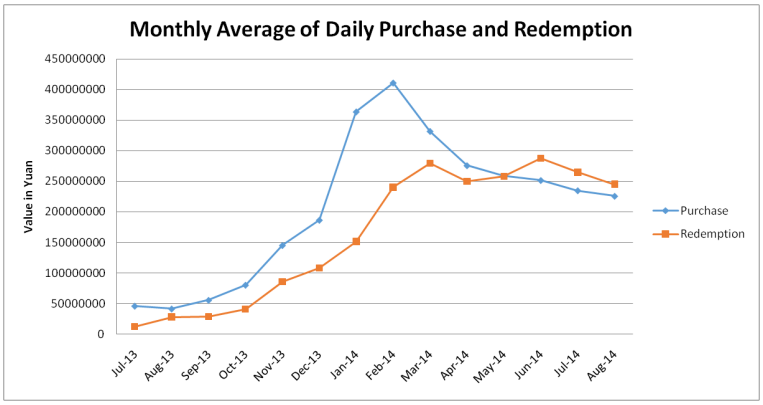


Fig.7 Monthly Average of Daily Purchase and Redemption

The source data in the 13 months given have significantly irregular fluctuations. This work selected the relatively stationary third interval with obvious patterns for analysis. Intuitively, data is relatively stationary, but it still is not fully stable. Therefore, the accurate judgment should be made by stationarity test. This work used unit root test to find that the original data is not stationary, which means differential treatment is necessary:

$$x_t = X_t - X_{t-1}$$

After the first-order difference, the result of data processing is as shown in Fig. 8.

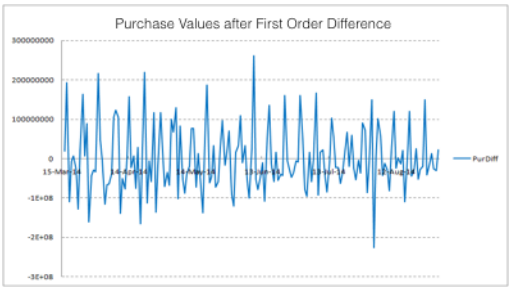


Fig.8(a) Purchase Values after First Order Difference

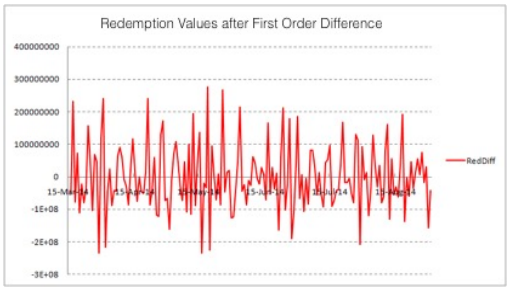


Fig.8(b) Redemption Values after First Order Difference

According to the quantitative observation on sequence chart, data seems to be generally stationary. Then, unit root test further proves that the present data has satisfied stationarity. After second-order difference, the stationarity obtained by unit root test decreases, and the actual score is also lower. Then, appropriate model choice is conducted according to the table.

	Autocorrelation function (ACF)	Partial autocorrelation function (PACF)
AR model	Trailing	Truncation
MA model	Truncation	Trailing
ARMA model	Trailing	Trailing

Table 2 ACF and PACF Characteristics and Appropriate Model Table

After drawing the autocorrelation function and partial correlation function after the first difference, ACF and PACF images were trailing according to the observation. Therefore, ARIMA model was the most suitable in analysis on this problem.

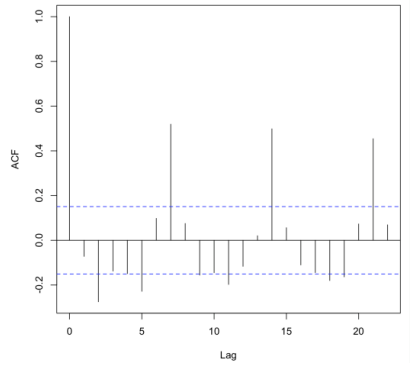


Fig. 9(a) ACF after Purchase Difference

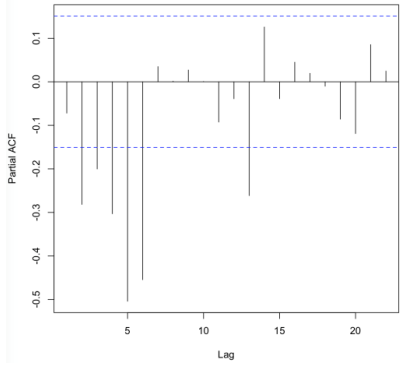


Fig. 9(b) PACF after Purchase Difference

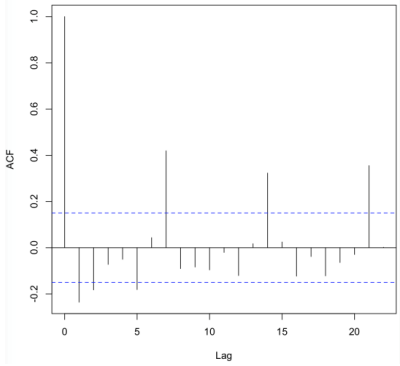


Fig. 9(c) after Redemption Difference

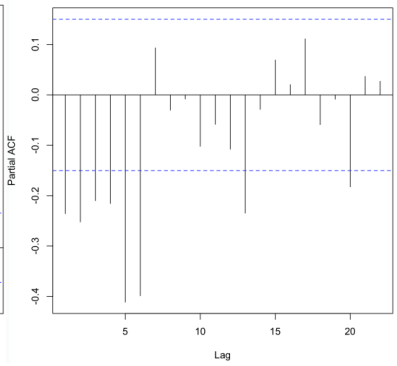


Fig. 9(d) PACF after Redemption Difference

6.1.3 Parameter Choice

ARIMA model includes three parameters, p, d and q, where p is the order of autoregression, q the moving average, and d the order of difference performed on time series to obtain stationarity.

According to the data stationarity process in the step above, d in this problem should be 1. This work gave the estimated value to p and q according to the periodicity of ACF and PACF images. Then, every group of p and q in possible range was exhausted, and every group of model was estimated according to Akaike Information Criterion and ranked by AIC value.

The results are listed in Table 3.

Ranking	p	d	q	AIC
1	9	1	9	4101.469
2	9	1	8	4102.527
3	10	1	9	4103.305
4	8	1	8	4103.426
5	8	1	9	4103.498
6	10	1	8	4103.581
7	5	1	5	4105.604
8	4	1	6	4105.871
9	10	1	10	4105.893
10	7	1	5	4106.276

Table.3(a) 10 Models with Minimum AIC in Purchase prediction

Ranking	p	d	q	AIC
1	5	1	5	4044.936
2	6	1	6	4047.057
3	5	1	8	4048.491
4	4	1	9	4049.639
5	7	1	6	4049.85
6	5	1	7	4050.602
7	8	1	6	4050.895
8	7	1	7	4050.919
9	7	1	8	4050.971
10	9	1	6	4051.175

Table.3(b) 10 Models with Minimum AIC in Redemption Prediction

Data in August 2014 are used as testing set. Models with each configuration are used to predict the total purchase redemption in August 2014. After comparing prediction value and actual value, the best ten models are selected according to the relative error of single data point. The results are listed in Table 4.

Ranking	p	d	q	Error
1	9	1	6	4.75807
2	7	1	8	4.912709
3	8	1	6	4.954694
4	6	1	6	4.97219
5	8	1	9	5.06238
6	7	1	5	5.236657
7	5	1	8	5.310632
8	4	1	8	5.520893
9	4	1	10	5.534177
10	7	1	6	5.553243

Table.4(a) 10 Models with best performance in Purchase prediction on Aug test set

Ranking	p	d	q	Error
1	7	1	10	7.918704
2	10	1	6	7.940052
3	10	1	10	7.972299
4	5	1	8	8.092522
5	3	1	10	8.248009
6	7	1	6	8.26122
7	9	1	5	8.321216
8	1	1	8	8.398347
9	1	1	9	8.497571
10	8	1	10	8.572706

Table.4(b) 10 Models with best performance in Redemption Prediction on Aug test set

6.1.4 Results and Analysis

Taking comprehensive consideration of Akaike Information Criterion and the estimation on ARIMA model with training data, the best model of purchase use is ARIMA (9, 1, 6) and that of redemption use is ARIMA (10, 1, 10). Fig. 10 and Table 5 show the prediction results and actual scores, respectively.

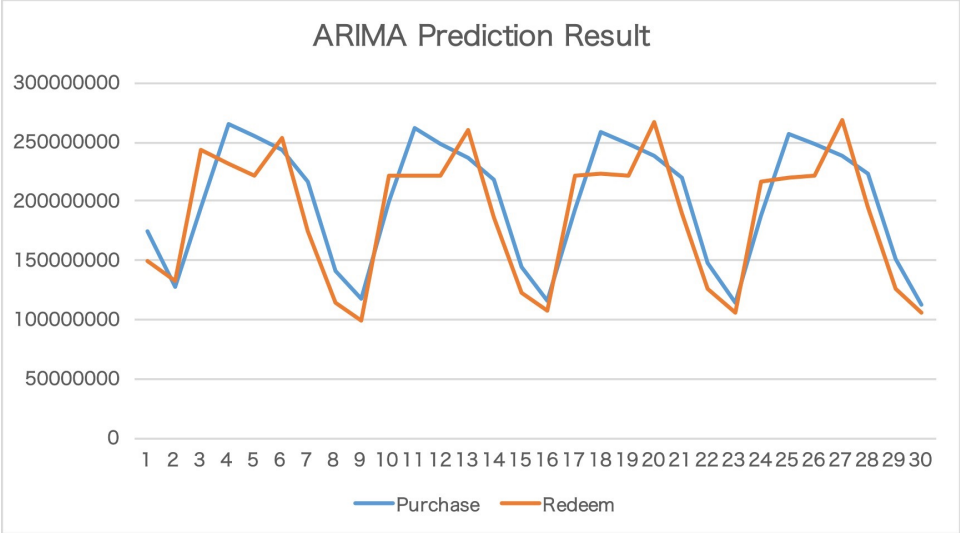


Fig.10 ARIMA Prediction Result

Purchase parameter	Redemption parameter	Score
ARIMA(7,1,4)	ARIMA(7,1,4)	80.38
ARIMA(7,1,5)	ARIMA(7,1,5)	82.23
ARIMA(7,2,5)	ARIMA(7,2,5)	78.02
ARIMA(7,1,6)	ARIMA(7,1,6)	79.96
ARIMA(7,1,7)	ARIMA(7,1,7)	78.41
ARIMA(6,1,6)	ARIMA(10,1,10)	84.21
ARIMA(9,1,6)	**ARIMA(10,1,10)**	**86.32**
ARIMA(9,1,6)	ARIMA(7,1,10)	83.74
ARIMA(5,1,8)	ARIMA(9,1,9)	83.22
ARIMA(5,1,8)	ARIMA(10,1,10)	83.86
ARIMA(7,1,8)	ARIMA(8,1,9)	84.39

Table 5 ARIMA Model Scores

As a typical analysis method for time series, ARIMA can calculate desirable prediction result by auto-regression and moving average method. This specific problem itself asks for prediction in unit of day – in the setting of a time series, suitable to use ARIMA models. This method can grasp important time characteristics well, and the large volume of data available can be used to train a relatively accurate model.

However, ARIMA model requires higher stationarity of data. The data in this subject cannot directly satisfy the requirements, therefore stationarity process must be conducted. Meanwhile, this model has high requirement on parameter adjustment, and the model is only effective with parameters. Generally speaking, ARIMA model is a advisable model with wide application, but its feasibility in this specific problem remains questionable.

6.2 Linear Regression Model (Non-autoregressive)

6.2.1 Model Introduction

As a regression algorithm widely applied in statistics and machine learning, linear regression can be used to determine the quantitative relationships between multiple variables. Given independent variables and dependent variables of a training data set, this model can perform iterations to set appropriate weight for every variable and fit the function. Therefore, when a new set of independent variables are input, values of dependent variables can be calculated with the trained linear relationship.

By treating this problem as a regression problem, some temporal features can be extracted for every day to be used with linear regression model. Then, training model can make prediction with these features as independent variables and the total purchase redemptions in the same day as dependent variables.

The nature of this model can be simply described as follows:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + +\theta_n x_n$$

where x_i represents independent variable and θ_i the weight of this variable that is, the coefficient.

Meanwhile, the aggregated difference between model prediction and actual values (training target) can be represented by Cost Function. In this work, least square cost function is used:

$$J(\theta) = J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - t^{(i)})^2$$

where $x^{(i)}$ is the independent variables of the ith sample and $t^{(i)}$ the target value of it. Then, following gradient descent is used to minimize the cost function step by step.

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - t^{(i)}\right)^2 x_j^{(i)} \quad (j = 0,1,\dots,n)$$

where α is learning rate that is, the step size in gradient descent. With negative gradient direction of function (the negative direction of partial derivative for every variable) as search direction, gradient descent updates parameters step by step until convergence.

6.2.2 Results and Analysis

Feature	Score
1	104.78
12	106.52
13	96.32
125	108.29
135	108.20
1245	**110.94**
124567	109.53

Table 6 Linear Regression Scores

Feature	Number
Day of the Week	1
Month(1/3,2/3,3/3)	2
Day of the Month	3
Day in Holiday	4
Holiday or Not	5
Moth Start	6
Month End	7

Table 7 Features Table

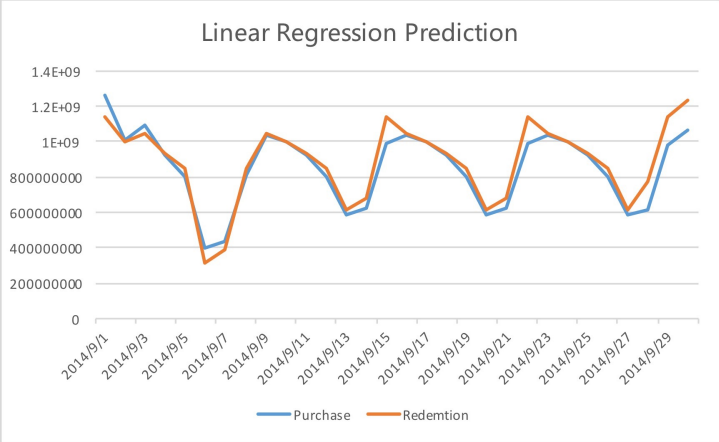


Fig. 11 Prediction Results of Linear Regression

Linear regression is an algorithm with simple concept which is convenient and useful in practical problems. It can accurately grasp the most important features by manual selection,

so it has better performance than time series prediction methods such as ARIMA in this problem. With proper feature selection and addition, this algorithm can be very effective. Unlike time series prediction, however, regression (non-autoregression) cannot analyze time series in a timeline and the relationship between consecutive values, which means some important information would be dropped. Moreover, for the month we need to predict, finance-related features are all unknown, so non-temporal features cannot be extracted effectively. This limitation restricts the model in utilizing the provided interest and yield rates. In actual evaluations, the results obtained by this method are not very satisfactory.
In addition, linear regression is prone to be affected by anomalous points, therefore its performance is inferior to GBDT regression in actual use.

7. Neural Network Model

7.1 Model Introduction

7.1.1 Model Overview

Neural network is a mathematic model for distributed parallel information processing algorithm based on the network behavior of animal nerves. Dependent on mass data input, this machine learning model can adjust the connections among internal nodes and use weight adjustment and function calculation to make prediction.
Composed of connected neurons, neural network can construct the limitation on every neuron with known data, and learn and save a large amount of mapping relations of input-output model. Moreover, this network can transform activation function by design according to demands to simulate the interaction among neurons. Therefore, the current neuron can activate the related neuron and, finally, the output neuron to output the predicted value.
In prediction problems, a commonly used neural network model is BP (Back Propagation) neural network, as known as the feedforward neural network. This network will pass the error reversely in calculation to input layer. Therefore, the parameter at every joint can be adjusted dynamically to make the whole model to the optimum by training.

7.1.2 Network Structure and Formula

The overall structure of neural network can be divided into input layer, hidden layer and output layer. Hidden layer may be composed of multiple layers.

Fig. 12 shows a three-layer feedforward network model, including input layer, hidden layer and output layer:

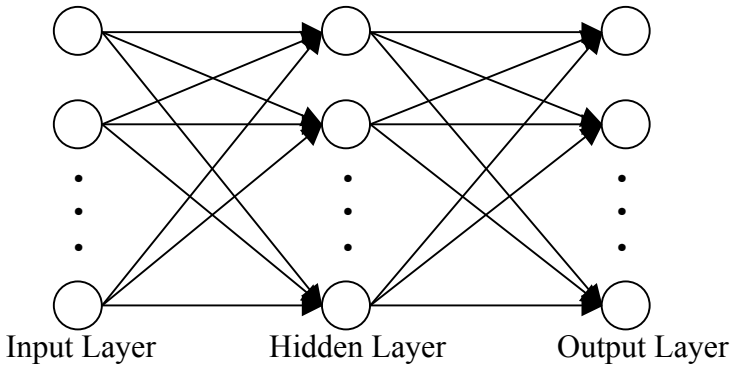


Fig. 12 Structure of Neural Network

The function expression of every hidden layer in BP neural network is as follows:

$$out_i = \sum_{j=1}^{k} w_j h_j$$

where out_i is the total output in ith layer, k the neuron number in hidden layer, w_j the weight of the output by jth neuron.

In other words, when the value in one hidden layer is transferred to the next layer, the output of next layer can be obtained by accumulating every neuron in this hidden layer with weight. This output is distributed to all neurons with initial weights in the next hidden layer for parallel processing according to the function for each neuron. Therefore, neural network can be viewed as multiple functions in serial connection after parallel processing. It can be used to optimize the calculation results by adjusting the structure of and weights in the network.

7.1.3 Neuron Structure and Expression

Fig. 13 shows the single node in network that is, the basic model of neuron.

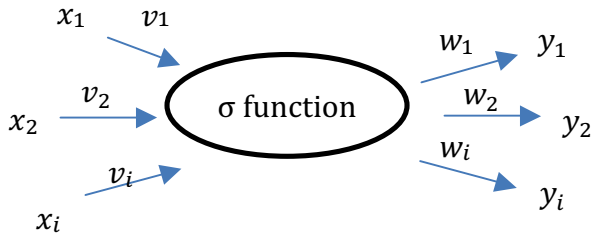


Fig. 13 Basic Model of Neuron

The function expression of every neuron in hidden layer is as follows:

$$h_j = \sigma(v_j \cdot \phi(x))$$

where $\phi(x)$ is the data input from the last layer, v_j the weight of the jth input neuron, and $\sigma(z)$ the Sigmond function which is the activation function of neural network. This work uses common Logistic function which has following expression:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

7.1.4 Back Propagation Process

In order to achieve the optimal configuration of neural network, the difference between predicted values by neural network and training targets are constantly calculated in training process. The error is then back propagated through hidden layer until it reaches input layer to distribute the error to all the neurons in every layer. Then the weight connecting every two nodes is corrected by gradient descent to minimize the cost function.

The cost function used in this paper is square error. The error of the ith sample is as follows:

$$E = \frac{1}{2}(t_i - y_i)^2$$

where E is the error, t_i the expected output, and y_i the actual output. $\frac{1}{2}$ is the 2 produced to eliminate the partial derivation.

By Chain Rule, the partial derivative of square error to every coefficient is obtained by following formula:

$$\frac{\partial E}{\partial v_{ij}} = \delta_j out_i$$

where out_i is the total output of ith neuron, v_{ij} the weight between ith and jth neurons, and $\delta_j = \frac{\partial E}{\partial out_j}\frac{\partial out_j}{\partial in_j}$ (in_j refers to the total input of jth neuron (the weight sum of all outputs in all preceding layers))

Therefore, the weight update in every step of iteration is as follows:

$$\Delta w_{ij} = -\alpha \frac{\partial E}{\partial w_{ij}}$$

α is Learning Rate that is, the size of each step in gradient descent.

The iteration stopping criteria in this work is that the gradient reaches 1×10^{-7} or the number of iteration reaches 1000. After achieving the required accuracy, all the nodes in the neural

network have been adjusted to the optimal status, iteration stops, and the trained model can be used to make prediction.

7.2 Parameter Selection and Feature Addition

In order to establish this prediction model, this work chooses the neural network function in Matlab and SAS (Statistical Analysis System). Besides the choice of features, the number of hidden layer neurons needs to be determined. This number should be set to ensure that the network properly fit the weight of every variable. The larger the number of hidden layer neurons is, the better the reflection of series trend and the better fit of the function. However, over fitting also becomes easier to occur. Generally, the value of this parameter can vary from several to dozens. The optimal parameter in this paper is obtained by trials and tests. According to the features of this subject, this work used Bayesian Regulation which has advantages when processing small-scale and noisy data for training. Following results and optimal parameters can be obtained after a large amount of trials and tests on parameters.

7.3 Results and Analysis

Purchase (No of Delays, No of Neurons)	Redemption (No of Delays, No of Neurons)	Score
2,2	2,2	66.82
7,2	7,2	80.27
7,6	7,6	89.09
7,8	7,8	90.57
7,8	7,5	90.99
7,10	7,5	91.2
7,12	7,5	90.84
6,10	6,5	87.23
6,8	6,5	88.02
8,10	8,5	92.17
8,10	**8,8**	**93.04**
9,10	9,8	91.29
9,8	9,5	90.93

Table 8 Autoregressive Neural Network Scores

Feature	Score
1	109.37
12	110.64
13	94.47
125	**114.36**
135	113.39
1245	114.21
124567	112.4

Table 9 Non-autoregressive Neural Network Scores

Feature	Number
Day of the Week	1
Month(1/3,2/3,3/3)	2
Day of the Month	3
Day in Holiday	4
Holiday or Not	5
Moth Start	6
Month End	7

Table 10 Features Table

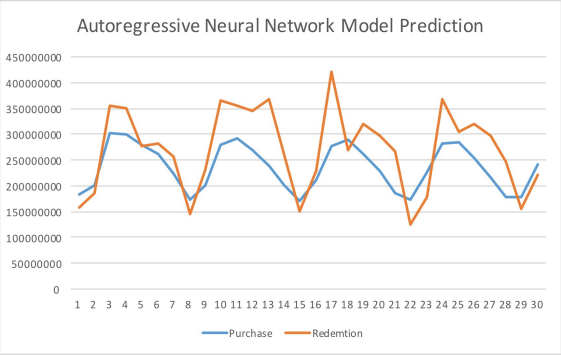


Fig.14 Autoregressive Neural Network Model Prediction

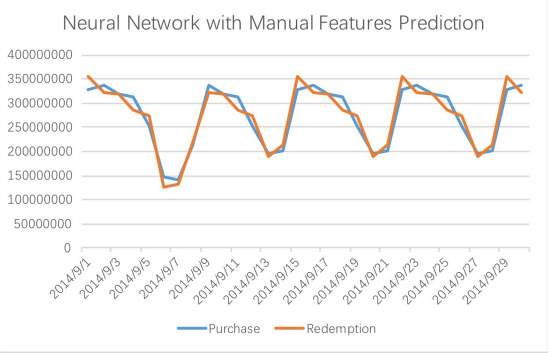


Fig.15 Neural Network with Manual Features Prediction

In fact, BP neural network realizes a mapping from input to output. It possesses good self-learning ability and is highly adaptive. In other words, as long as enough data is provided, neural network model can make accurate predictions based on the mapping.

However, neural network model does have drawbacks. Firstly, extension from partial to overall is used in the process of weight adjustment and error minimization. This method may cause partial weight minimization and failure of the network training. Secondly, a lot of neural network structures are available, while there is no complete theoretical system on how to choose between them. Most determination methods are still arbitrary. Therefore, the effectiveness of prediction model is questionable. Thirdly, although the number of hidden layer neurons can be adjusted, over fitting frequently occurs, which leads to a decrease in prediction ability with the increase of accuracy of fitting training data, nullifying the model in actual use.

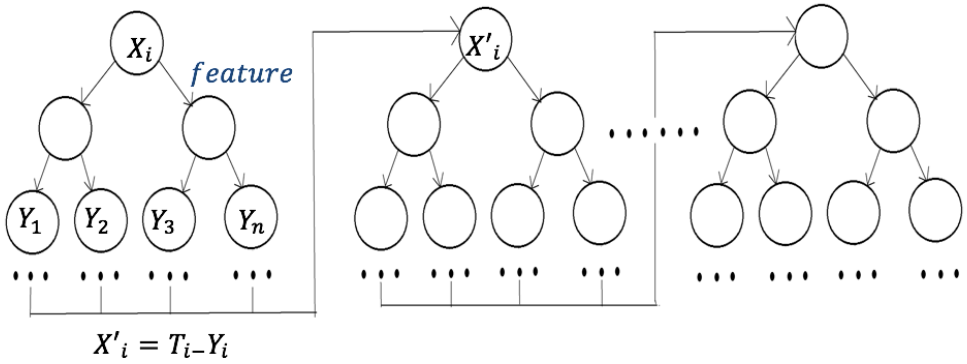
8. GBDT Regression Model

8.1 Model Introduction

GBDT (Gradient Boosting Decision Tree) is an iterative decision tree algorithm composed of multiple decision tress. The final predicted value is made by accumulating all the output of decision trees. This model is mainly composed of three factors: decision tree, gradient descent and step length.

8.1.1 Construction of Single Decision Tree

Construction of single decision trees is the basis of constructing the GBDT model. In this work, decision trees used are binary decision trees. The number of features and leaf nodes in every decision tree is certain. The features selected for use in decision trees in this work include day of the week, day of the month, etc. Starting from the root, in each iteration the attribute with the most Information Gain is selected, and the set is split by that attribute.



Then this process recurs on each subset, until a decision tree is built.

Fig. 16 shows the overall structure of GBDT model.

Fig. 16 Schematic Diagram of GBDT Decision Trees

In above figure, X_i represents the input of the ith sample, Y_i the output of the ith sample, T_i the target value of the ith sample that is, the expected output. The connection between two nodes is always completed by whether a sample satisfies the classification criterion.

In order to determine the features at every branch, information entropy is used:

$$H = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$$

where n means that there are n attributes, each can be used to divide samples into two categories. The probabilities p_1 and p_2 of these two categories occurring in each sample are calculated, and information entropy of the unselected attributes before branching is calculated by above formula with the probabilities. Then, attribute x_i is selected for branching according to following rules: if x_i has this feature, the sample is classified to one branch; if no, put the sample to another branch. Then, the entropies of these two branches H_1 and H_2 are calculated. The total information entropy after branching is calculated as follows: $H' = p_1 \times H_1 + p_2 \times H_2$, and the information gain at this time is $\Delta H = H - H'$. All the attributes are tested to determine the final selection. Only the attribute with the largest information gain is selected as the attribute of this branching.

After the construction of decision tree has been completed, the current data series is input into decision tree at root node for classification. All the data series will reach a leaf node, and the value at every leaf value is assigned and as the average value of all samples at this node. The formula for the value at the kth leaf node, W_k is as follows:

$$W_k = \frac{\sum_{i=1}^{t_k} x_i}{t_k}$$

where t_k is the total number of samples reaching the kth leaf node and x_i the value of every sample.

After finishing the training for decision tree system, the input sample to be predicted will reach to one leaf node through the classification of decision tree according to features. The value at this leaf node will be the result obtained by data in current decision tree.

8.1.2 Gradient Iteration (Establishment of Connections Between Trees)

During the training of GBDT model, the previous decision tree will transfer the error between predicted value and target value of every sample to the root node of next decision tree as target value. Then, the same sample input is used to fit with feature to realize the cooperated serial processing among multiple trees.

$$x_j^{(i)} = x_{j-1}^{(i)}$$

$$t_j^{(i)} = t_{j-1}^{(i)} - y_{j-1}^{(i)}$$

$x_j^{(i)}$ is the input of ith sample in jth tree, $t_j^{(i)}$ the target output of ith sample in jth tree, and $y_j^{(i)}$ the actual output of ith sample in jth tree.

When the cost function reaches a threshold, iteration stops, and the training of GBDT model is done.

8.1.3 Output Calculation

When data to be predicted pass through all the decision trees in GBDT models and obtain one result from every decision tree, the final prediction value of ith sample is the sum of the result values from all decision trees.

$$W_i = \sum_{j=1}^{n} y_j^{(i)}$$

where n is the number of decision trees and $y_j^{(i)}$ the output result of ith sample in jth tree.

8.2 Results and Analysis

No of Trees	No of Delays	Score
400	7	92.15
800	7	95.02
1000	7	99.84
1100	**7**	**102.66**
1200	7	101.13
1300	7	98.3
1100	8	100.14
1100	6	99.53
1100	5	100.07

Table 11 Autoregressive GBDT Regression Scores

Feature	Score
1	109.58
12	111.39
13	101.46
125	114.29
135	113.97
1245	**115.01**
124567	114.4

Table 12 Non-autoregressive GBDT Scores

Feature	Number
Day of the Week	1
Month(1/3,2/3,3/3)	2
Day of the Month	3
Day in Holiday	4
Holiday or Not	5
Moth Start	6
Month End	7

Table 13 Features Table

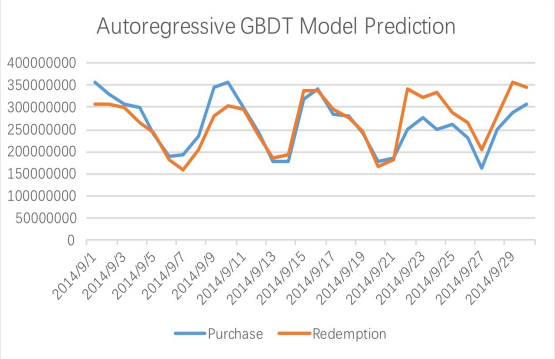


Fig.17 Autoregressive GBDT Model Prediction

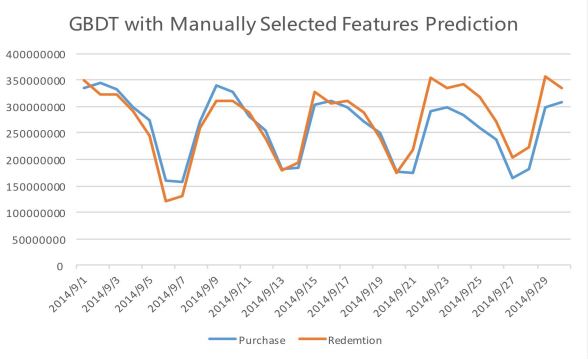


Fig.18 GBDT with Manually Selected Features Prediction

GBDT model is suitable for almost all linear or nonlinear regression problems and can be applied to all the scenes. It is a relatively perfect algorithm with advantages of both linear regression and decision tree. With the joint judgment by multiple decision trees, the errors of each data point can be minimized effectively to achieve a good prediction accuracy. This work added appropriate features into GBDT by data observation, correlation test and actual evaluation. From those tests, a good model grasping relationship between variables was obtained. In final evaluation, this algorithm obtained a high score.

9. Model Ensemble

Based on the three original models and six methods (each model includes autoregressive and non-autoregressive methods), this work further experiments with model ensemble. This work selected non-autoregressive Neural Network (NN) and non-autoregressive GBDT

regression algorithms which performed the best among the methods, and combined the prediction of two models at each data point to predict according to the following formula:

$$y_i^{(final)} = \theta \cdot y_i^{(GBDT)} + (1-\theta) \cdot y_i^{(NN)}$$

Table 14 shows the obtained results.

θ Value	Score
0.9	116.43
0.7	**117.25**
0.5	115.69

Table 14 Model Ensemble Scores

According to the table above, with appropriate use, model ensemble can improve the overall prediction accuracy to some extent. However, the prediction accuracy may decrease with random parameter setting. Fig. 19 shows the best ensemble method which improves the overall prediction accuracy in final evaluation.

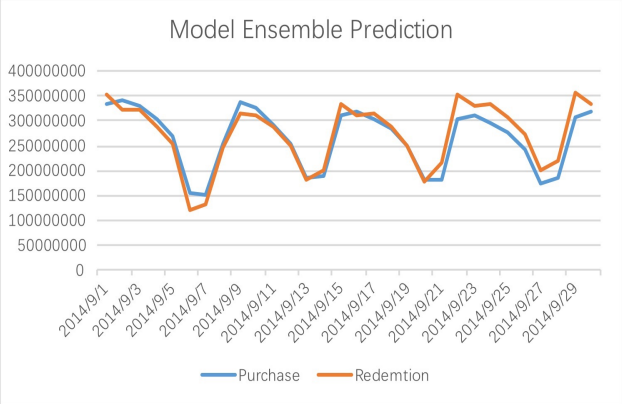


Fig. 19 Model Ensemble Prediction

Generally, model ensemble is a method worth trying because it has potential to combine various models and improve prediction accuracy. However, there are different answers to how to ensemble models and how to select the ratio in every specific application. Therefore, large numbers of trials is necessary in practical use.

10. Conclusions and Prospects

10.1 Final Prediction Results

Model		Highest Score
Linear	autoregressive	86.32
	non-autoregressive	110.94
Neural Network	autoregressive	93.04
	non-autoregressive	114.36
GBDT	autoregressive	102.66
	non-autoregressive	115.01
Ensemble	**NN+GBDT**	**117.25**

Table 15 Comparison of Highest Score by Different Methods

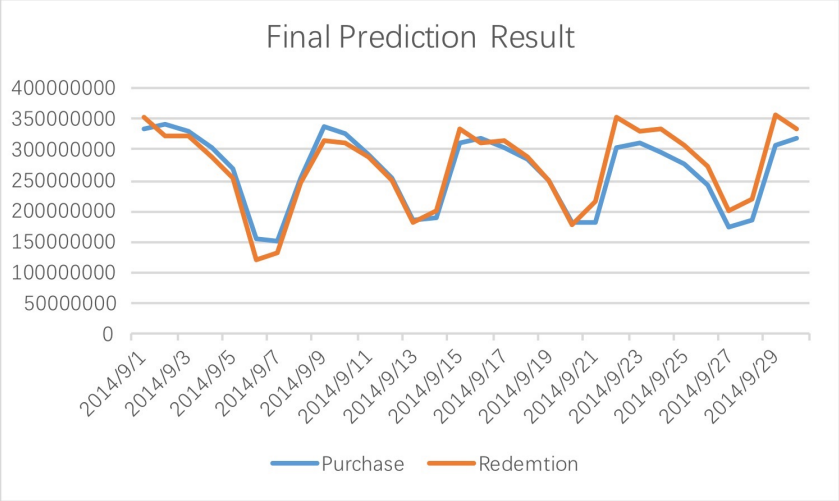


Fig. 20 Final Prediction Results

Table 15 shows the comparison of the best results by five prediction methods in this work, among them model ensemble achieved the highest score. Fig. 20 is the final prediction results of this work which is obtained by ensembling GBDT Regression model and Neural Network model. The actual daily purchase and redemption values in September 2014 has not been publicized, and a undisclosed nonlinear function is used in official evaluation carried out by Alibaba (see Section 2.2). Therefore, scores can only be used to compare the relative superiority of results, rather than tracing back to quantitative prediction accuracy. In the competition on the same subject by Alibaba, the author Chunyang Xie in this work passed two rounds of elimination, and reached a final rank of 193rd place among 4,868

participating teams who are all college undergraduate and graduate students. Therefore, the prediction results introduced in this paper should be relatively accurate.

10.2 Summary of Prediction Methods

For similar prediction problems of predicting capital flow, GBDT regression works the best with properly selected features. It may be because GBDT Regression is nonlinear and has broader adaptability. Moreover, the joint collaboration of multiple trees can minimize error in each sample effectively. After establishing regression formula, this work analyzed and selected features with originality, including day of the week, day of the month and holiday etc., and measured the correlation between them and the variables to predict. Finally, features of week, month and holiday with high correlation are added in the model as binary flag. However, this result is not absolute. The performance of every model may vary with different sets of data and different scenarios.

10.3 Deficiencies and Prospects

In this research work, the use of GBDT regression model neglects temporal continuity of the data and trend on data development, while those are important features that should be considered in data analysis. Model ensemble in Chapter 9 is only a simple trial. In the future, more model ensemble methods will be experimented to combine advantages of different models and to improve the prediction accuracy.

In addition, the sample size of 300,000 users in this work is relatively small compared with 1,850,000 users of Yu'EBao. There may be some differences in sampling methods for random factors, so it is uncertain to completely reflect the rule for cash flow of Yu'EBao. In other words, the result may be different when using these methods in another user set or the universal set.

Reference

1.Wang, Songgui. Xian Xing Hui Gui Xi Tong De Yi Zhong Xin De Gu Ji, Gong Chen Shu Xue Xue Bao, 1998. Print

2.Wang, Huiwen, And Jie Meng. Duo Yuan Xian Xing Hui Gui De Yu Ce Jian Mo Fang Fa, Beijing Hang Kong Da Xue Xue Bao, 2007. Print

3.Fu, Gong, And Chen Yun. Ying Yong Xian Xing Hui Gui Fen Xi, Zhong Guo Ren Min Da Xue Chu Ban She, 1989. Print.

4.Liu, Xiaohu, And Sheng Li. Jue Ce Shu De You Hua Suan Fa, Ruan Jian Xue Bao, 1998. Print.

5. Tang, Huasong, And Huiwei Yao. Shu Ju Wa Jue Zhong Jue Ce Shu Suan Fa De Tan Tao, Ji Suan Ji Ying Yong Yan Jiu, 2011. Print.

6.Durking J, Jingfei Cai, And Zixing Cai. Ju Ce Shu Ji Shu Ji Qi Dang Qian Yan Jiu Fang Xiang, Kong Zhi Gong Cheng, 2005. Print.

7.Tian, Miaomiao. Shu Ju Wa Jue Zhi Jue Ce Shu Fang Fa Gai Shu, Chang Chen Da Xue Xue Bao, 2005. Print.

8. Tan, Xu, And Liling Wang. Li Yong Jue Ce Shu Fa Jue Fen Lei Gui Ze De Suan Fa Yan Jiu, Yun Nan Da Xue Xue Bao, 2000. Print.

9.Ma, Xiuhong, Jianshe Song And Shenfei Dong. Shu Ju Wa Jue Zhong Jue Ce Shu De Tan Tao, Ji Suan Ji Gong Cheng Yu Ying Yong, 2004. Print.

10. Li, Cheng. Shen Jing Wang Luo Xi Tong Li Lun, Xi An Dian Zi Ke Ji Da Xue Chu Ban She, 1900. Print.

11. Li, Qun. Ren Gong Shen Jin Wang Luo Li Lun, She Ji Yu Ying Yong, Hua Xue Gong Ye Chu Ban She, 2007. Print.

12.Hegen, Demusi, Bier. Shen Jin Wang Luo She Ji, Ji Xie Gong Ye Chu Ban She, 2002. Print.

13. Wang, Chunhui, And Haihui Wang. Ji Yu Shen Jing Wang Luo Ji Shu De Shang Ye Yin Hang Xin Yong Fen Xian Ping Gu. Xi Tong Gong Chen Yu Shi Jian, 1999. Print.

14. Jiao, Licheng. Shen Jing Wang Luo De Ying Yong Yu Shi Xian. Xi An Dian Zi Ke Ji Da Xue Chu Ban She, 1993. Print.

15. Gong, Guoyong. ARIMA Mo Xing Zai Shen Zhen GDP Yu Ce Zhong De Ying Yong. Shu Xue De Shi Jian Yu Ren Shi, 2008. Print.

16. Xiong, Zhibing. Ji Yu ARIMA Yu Shen Jing Wang Luo Ji Cheng De GDP Shi Jian Xu Lie Yu Ce Yan Jiu, Shu Li Tong Ji Yu Guan Li, 2011. Print.

17. Sun, Caiyun And Xiaojing, Yang. Cheng Ji ARIMA Mo Xing De Jian Li Ji Ying Yong, Hua Bei Ke Ji Xue Yuan Xue Bao, 2008. Print.

18. Li, Qiaomei And Guojing, Xiong. She Hui Xiao Fei Pin Ling Shou Zong E ARIMA Mo Xing De Jian Li Ji Yu Ce. Ke Ji Guan Chang, 2007. Print.

19. Chang, Liang. Ji Yu Shi Jian Xu Lie Fen Xi De ARIMA Mo Xing Fen Xi Ji Yu Ce, Ji Suan Ji Shi Dai, 2011. Print.

20.Schonlau M. Boosted Regression (Boosting): An Introductory Tutorial And A Stata Plugin. The Stata Journal, 5(3), 330-354.