Research on Random Forest Algorithm Based on Big Data in Parallel Load Forecasting

Qingqing Liu

Luoyang Normal College, Luoyang Henan 471934 China

Abstract. The paper propose a parallel load forecasting method based on random forest algorithm, through the analysis of historical load, temperature, wind speed and other data, the algorithm can shorten the load forecasting time and improve the processing capability of large data. This paper also designs and implements parallel load forecasting prototype system based on power user side large data of a Hadoop, including data cluster management, data management, prediction classification algorithm library and other functions. The experimental results show that the accuracy of parallel stochastic forest algorithm is obviously higher than decision tree, and the prediction accuracy on the different data sets is generally higher than decision tree, and it can better analyze and process large data.

Keywords: Power user side; Parallel processing; Load forecasting; Large data;

1 Introduction

Power load forecasting plays a very important role in ensuring power system planning and reliable and economic operation. With the emergence of intelligent power massive data, we must find a new method to meet the requirements of mass data analysis. The existing prediction algorithms cannot meet the requirements of prediction speed and prediction accuracy. The traditional local weighted linear regression prediction has the advantages of fast training and low prediction error rate. Because the algorithm needs to find a close neighbor for each test point, the amount of computation is very large for big data, and the time can reach several hours or days by single computer operation. Therefore, it is very important to solve the prediction problem based on massive data.

The random forest is an integrated learning method, taking decision tree as the basic learning unit, including a number of decision trees trained by Bagging integrated learning theory and random subspace method, input the samples to be classified, and generate the result by each decision tree, and the final classification result is cast by the result of each decision tree. Random forest is an integrated learning method of multiple decision trees. It can not only overcome some shortcomings of decision tree, but also have good scalability and parallelism. It can effectively solve the fast processing problem of large data. It has a good application prospect for power load forecasting in large data environment.

2 The principle of random forest algorithm

Figure 1. Training process of single decision tree in random forest.

The random forest [1-3] is made up of a series of classified regression trees. In 2001, it was proposed by Leo Breiman based on his Bagging integrated learning theory and the random subspace theory proposed by Ho. In the random forest, each classification regression tree has its own independent sample training set TS, and TS is composed of samples extracted by Bagging algorithm from the total sample S and S and so on. The algorithm uses each TS to train the classification regression tree. In the process of forming each classifier, the branch of each internal node selects several attribute values randomly according to the random subspace theory, and finally forms a decision tree

with classification rule or regression function. The final result of random forest is the vote selection for each classification tree, or the average of the results of each classification tree. The training process of a single decision tree in random forest is shown in Figure 1.

The construction process of a single classification regression tree mainly consists of selecting the appropriate attribute values from the attribute set to branch, and then repeating the search process repeatedly on the subtrees produced by them until a stop growth rule is met. The selection of the branch attribute values is based on the Gini index and the least square deviation, in which the Gini index is suitable for the classification tree, and the least square deviation is suitable for the regression tree, and the concrete calculation is as follows:

1) Gini index.

The Gini index can measure the impure nature of nodes.

$$GIN(T) = 1 - \sum_j p^2(j/t) \quad (1)$$

Where: T is the branch attribute of the current node; P (j/t) represents the ratio of the target class J in the node T. The node T is defined by the Gini criterion based on the attribute value s:

$$GINI(s, t) = p_L GINI(t_L) + p_R GINI(t_R) \quad (2)$$

The partition standard is to make the GINI (s, t) minimum.

2) Least square deviation.

The least squares deviation is used to measure the regression tree, and the fitting error formula of node T is:

$$Err(t) = \frac{1}{n_t} \sum_{D_t} (y_i - k_t)^2 \quad (3)$$

Where: NT is the number of instances in the node T; KT is the average value of the target value for each instance in the node.

$$k_t = \frac{1}{n_t} \sum_{D_t} y_i \quad (4)$$

The least square deviation standard divided by the attribute value s for node t is defined as:

$$Err(s, t) = \frac{n_{tL}}{n_t} Err(t_L) + \frac{n_{tR}}{n_t} Err(t_R) + C \quad (5)$$

In order to simplify the computation process in computer and avoid repeatedly traversing the attribute values, the formula (6) is simplified as:

$$Err(s, t) = \frac{S_L^2}{n_{tL}} + \frac{S_R^2}{n_{tR}} + C \quad (6)$$

Wherein: $S_L = \sum_{D_{tL}} y_i, S_R = \sum_{D_{tR}} y_i$, the divided standard is the max value of formula (6).

3 Load forecasting process based on parallel random forest

3.1 Forecast process of parallel random forest short term load

The specific prediction process of the algorithm is shown in Figure 2. The whole model is built on the distributed cluster of Hadoop, which stores large data in a distributed way. Map Reduce is used to parallelize the random forest algorithm. The algorithm can rely on the storage and computing power of the Hadoop cluster to mine and calculate the data. The whole process is executed in parallel. It can effectively improve the accuracy of prediction and improve the ability of load forecasting system to deal with large data.

Figure 2. Load forecast flow chart based on parallel random forest in short term.

3.2 Evaluation index of load forecasting error

The prediction of power load is the estimation of future power load by historical data. There are many reasons to generate the error, which are summarized as follows: 1) the simplification of the mathematical model and the neglect of the relation of various factors; 2) the historical data is not complete; 3) the error is caused by the improper selection of the parameters.

The indicators used in this paper are as follows:

If $y(i)$ and $\hat{y}(i)$ represents the actual load and forecast value at i time respectively.

Absolute error: $E = y(i) - \hat{y}(i)$

Relative error:

$$e_1 = \frac{1}{n} \sum_{i=1}^n \left| \frac{y(i) - \hat{y}(i)}{y(i)} \right| \quad (7)$$

Where: e_1 is a daily mean error. Because the prediction error is positive and negative, in order to avoid the positive

and negative cancellation, the absolute value of error is calculated when the mean is calculated.

$$e_2 = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{|y(i) - \hat{y}(i)|^2}{y(i)^2}} \quad (8)$$

Where: E_2 is the root mean square error. The root mean square error index strengthens the error of numerical value and improves the sensitivity of index.

4 Load forecasting experiment and result analysis

The data source of this article is the load data and weather data collected by a power grid enterprise. The training data range is from November 24, 2011 to November 30, 2011. The interval of each device is 15min, as shown in Table 1. Forecast the power load in December 1, 2011, as shown in Table 2.

Table 1. Training data.

Date	Hour	Minute	High Temperature	Low Temperature	Quantity of electricity/kW.H
1	00	15	7	-3	52679.1854
2	00	30	7	-3	58145.652
...	...	...	...	...	...
9	23	30	3	-5	85263.4522
10	23	45	3	-5	74236.5584

Table 2. Actual load data on December 1, 2011.

Hour	Minute	High Temperature	Low Temperature	Quantity of electricity/kW.H
00	15	8	-1	53200.525
01	30	8	-1	53106.952
...	...	...	...	...
23	30	8	-1	72112.502
23	45	8	-1	71102.308

The influence of factors such as temperature, humidity, working day, holiday, season and other load factors on the load fluctuation of power users is considered. By calculating the load intensity, it can provide a basis for the establishment of more accurate load forecasting model. The data expressed in this paper are: load time sequence is $x_1, x_2, \dots, x_n$; (x_1, i) are load data; (x_2, i) is a temperature sequence; (x_3, i) is a humidity sequence and so on.

The comparison between load forecast and actual load is shown in Table 3. As shown in figures 3 and 4, the trend of predicted value curve is similar to actual value, and the mean square root error is 3.01%. The error of prediction result accords with the error standard of load forecasting.

Table 3. Comparison results.

No	Predicted value	Actual value	Err/%
1	50976.581	53100.606	-4.0
2	54255.233	53105.956	2.4
...	...	...	...
95	72365.255	70125.553	-2.5
96	73157.718	71100.308	3.4

Figure 3. Load forecasting contrast curve.

Figure 4. Mean square error curve of load forecasting.

5 Conclusion

Combined with the current research status of large power data at home and abroad, this paper analyzes the characteristics of user side big data, and puts forward a big data analysis platform, and develop the parallel power load forecasting prototype system based on Hadoop for power user side. On this prototype system, the parallel prediction experiment for load is carried out by the parallel algorithm of random forest. The experiment shows that the method improves the precision of load forecasting.

References

1. Ding Jie, Xi Houwei, Han Haiyun, et al. A smart grid-oriented data placement strategy for data-intensive cloud environment [J]. Automation of Electric Power Systems, 2012, 36 (12): 66-70 (In Chinese).
2. Cao Zijian, Lin Jin, Song Yonghua. Optimization model for resources allocation of cloud computations in active distribution networks [J]. Proceedings of the CSEE, 2014, 34 (19): 3043-3049 (In Chinese).
3. Zhang Shaomin, Li Xiaoqiang, Wang Baoyi. Design of data security storage in smart grid based on Hadoop [J]. Power System Protection and Control, 2013, 41 (14): 136-140 (In Chinese).
4. Wang Baoyi, Zhao Shuo, Zhang Shaomin. Distributed power load forecasting algorithm based on cloud computing and extreme learning machine [J]. Power System Technology, 2014, 38 (2): 526-531 (In Chinese).
5. Zhong Qing, Sun Wen, Yu Nanhua, et al. Load and power forecasting in active distribution network planning [J]. Proceedings of the CSEE, 2014, 34 (19): 3050-3056 (In Chinese).