2019 18th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Multi-dimensional Short-term Load Forecasting Based on XGBoost and Fireworks Algorithm

Guilong Suo, Liangcai Song, Yanmei Dou, Zhiyong Cui

Hebi Power Supply Company, State Grid Henan Electric Power Company

{27394156, 151493526}@qq.com, songlc1990@sohu.com, cuizhiyongc@126.com

Abstract- Nowadays, most load forecasting of power system only takes advantage of sequential characteristics of the load itself. In fact, meteorological conditions, population and geographical factors will also have a significant impact on short-term load, thus reducing the accuracy of load forecasting. In this paper, we introduce XGBoost into load forecasting, making full use of its parallelism, anti-overfitting, second-order Taylor expansion and other characteristics, and modeling with multi-dimensional factors. The results show that the XGBoost model considering only sequential characteristics works fine in predicting the trend of short-term load, while the multi-dimensional model combined with temperature, humidity and rainfall factors has a significant improvement in prediction accuracy.

Keywords-*XGBoost; load forecasting; fireworks algorithm*

I. Introduction

Short-term load forecasting plays a key role in dispatching of power plants. Considering that the load is affected by many factors, such as regional economic level, electricity price, population, electricity structure and meteorological conditions, these factors will lead to fluctuation of power grid load, thus interfere with load forecasting. In order to improve the accuracy of short-term load forecasting, we need to consider the impact of external factors other than historical load data.

Reference [3-5] introduces the traditional bus load forecasting method. Because there is no distributed power supply, the load can be predicted directly by sequential characteristics extrapolation method, and the results are corrected by taking into account the weather, user characteristics and other related factors, so as to improve the accuracy of forecasting. Li presents a short-term bus load forecasting method based on LSSVM and Markov chain[6]. In [7], PSO optimization method is introduced into the neural network to improve the accuracy of bus load forecasting.

To achieve more efficient and accurate load forecasting, this paper establishes a multi-dimensional short-term load forecasting model based on XGBoost. In order to avoid empirical and random selection of hyper parameters, the fireworks algorithm with global convergence and high efficiency is used to optimize the combination of hyper parameters. The model is trained and tested by load data of a power plant within a week, and is evaluated by MAE, RMSE, MAPE and other performance indicators.

II Short-term Load Forecasting Model Based on XGBoost and FWA

A. XGBoost Principle

XGBoost is a tree learning algorithm, which takes decision tree as the basic unit of the model. The decision forest composed of many decision trees is the final learning model of XGBoost. It tries to correct the residual of all the previous weak learners by adding new weak learners. When these learners are combined for final prediction, the accuracy will be higher than that of a single learner.

The calculation of XGBoost is as follows:

(1) Firstly, the sample weights and model parameters are initialized, all samples in training set are given the same weight.

(2) For each iteration, equation (1) is used to calculate the error rate of classification:

$$err_m = \frac{\sum w_i I\left(y_i \neq G_m x_i\right)}{\sum w_i} \tag{1}$$

where w_i is the weight of the i-th sample, G_m is the m-th classifier.

(3) Calculate $a_m = \log\left(\left(1-err_m\right)/err_m\right)$.

(4) In the $m+1$-th iteration, update the weight of the i-th sample as $w_i e^{a_m I\left(y_i \neq G_m x_i\right)}$.

XGBoost constructs an objective function and then tries to optimize it. By making a second-order Taylor expansion of the loss function and adding a whole regular term outside the objective function, we balance the decline of the objective function and the complexity of the model, so as to avoid over-fitting. The objective function is as follows:

$$Obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2}$$

where n is the dimension of feature vector, $L(y_i, \hat{y}_i)$ is loss function, y_i is true value, $\hat{y}_i$ is predicted value, $\Omega(f_k)$ is regular term, used to control the complexity of the tree structure.

In XGBoost, we define complexity as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{3}$$

where T is the number of leaf nodes, γ specifies the minimum loss function descent value of node splitting, the larger this parameter, the more conservative the algorithm. λ is the L2 regular term of weight. The square L2 modulus of w is used to control the complexity of the tree, and avoid overfitting.

B. Hyper Parameters Optimization Based on FWA

The selection of appropriate hyper parameters of XGBoost is directly related to the performance of the

2473-3636/19/$31.00 ©2019 IEEE
DOI 10.1109/DCABES48411.2019.00068

model, but there is no universal and scientific method to determine the hyper parameters. In order to reduce the randomness and blindness, based on the second search of multi-dimensional grids, a method of hyper parameters optimization is proposed. Each hyper parameters combination is attempted in a grid traversal manner in turn.

As a swarm intelligence optimization algorithm, fireworks algorithm obtains the global optimum by exploding search mechanism. The algorithm has a behavior similar to fireworks explosion to search the solution space of the target problem efficiently. It has good performance and efficiency in solving complex optimization problems. Through the observation of fireworks explosion in the sky, two results are gotten: the first is the fireworks with good explosion effect, which produce more sparks when exploding, and the sparks are concentrated at the center of the explosion; the other is the fireworks with poor explosion effect, which produce few sparks, and scattered far from the center of the explosion.

Each spark is like a set of hyper parameters in search space. In this paper, the prediction performance of the model is used as the criterion to evaluate the effect of fireworks explosion, good prediction performance indicates that the explosion effect of this set of hyper parameters is good, and it is certainly possible that there are other good solutions nearby. In this case, more sparks will be generated around the set, which is used to search the adjacent area to improve local search ability. The spark of the selected hyper parameters combination will become a new combination in the next iteration.

Suppose there are n hyper parameters combinations, $X_i = \{x_1, x_2, \mathrm{L}\ , x_d\}$ is a d-dimension hyper parameters combination in search space, $f(X_i)$ represents the value of fitness function in X_i, i.e. the prediction error of load forecasting model. The less this value, the higher the fitness. Get the number of sparks generated by each hyper parameter combination through equation (4):

$$S_i = m \cdot \frac{f_{\max} - f(X_i) + \xi}{\sum_{i=1}^{n} \left[f_{\max} - f(X_i) \right] + \xi} \tag{4}$$

where m controls the total number of sparks, ξ is a minimum value close to zero to avoid dividing zero errors. f_{max} is the prediction error of the hyper parameter combination with worst explosion effect, namely:

$$f_{\max} = \max\left\{ f(X_i) \right\}, i = 1, \mathrm{L}\ , n \tag{5}$$

To avoid generating too many sparks around combination with good performance and falling into local optimum, or too few sparks around the one with poor performance and missing the global optimum, two boundary constant parameters a,b ($0<a<b<1$) are defined specifically, thus the number of sparks generated by X_i is revised as:

$$S_i' = \begin{cases} \mathrm{round}(a \cdot m), S_i < a \cdot m \\ \mathrm{round}(b \cdot m), S_i > b \cdot m \\ \mathrm{round}(S_i), otherwise \end{cases} \tag{6}$$

Compared with combinations with poor explosion effect, the sparks of the ones with good explosion effect are large in number and more concentrated. The displacement amplitude of explosion of X_i is defined as follows:

$$A_i = A \cdot \frac{f(X_i) - f_{\min} + \xi}{\sum_{i=1}^{n} \left[f(X_i) - f_{\min} \right] + \xi} \tag{7}$$

where A is the preset largest explosion range, $f_{\min}$ is the fitness function value of the combination with the best explosion effect, i.e.:

$$f_{\min} = \min\left\{ f(X_i) \right\}, i = 1, \mathrm{L}\ , n \tag{8}$$

For the d-dimension solution space, z of the dimensions are randomly selected for displacement after explosion each time:

$$z = \mathrm{round}\left[d \cdot U(0,1) \right] \tag{9}$$

where $U(0,1)$ represents a random number that obeys a uniform distribution in $[0,1]$.

At the beginning, the fireworks algorithm randomly selects n locations as the combinations to explode. After the explosion, they and the sparks generated by them constitute a set *M*. The combination with the best prediction performance will always be selected into the next iteration. For the remaining $n-1$ combinations and sparks, to ensure diversity of combinations, we select the combination or spark far away from others into the next generation. The distance between combination or spark X_i and others can be calculated by equation (10):

$$D_i = \sum_{X_j \in M} \left| X_i, X_j \right| \tag{10}$$

The probability that X_i is selected into the next iteration is:

$$p(X_i) = \frac{D_i}{\sum_{X_j \in M} D_j} \tag{11}$$

The specific steps of optimization based on fireworks algorithm are as follows:

(1) Randomly select n positions in solution space, each of them is called a spark, represents a hyper parameters combination.

(2) For each combination, the fitness value, i.e. the prediction error of the model, is calculated. The smaller the fitness value, the better the performance of the hyper-parameter combination is.

(3) The number and amplitude of sparks generated by the combinations are determined according to the quality of them. The better the combination, the smaller the radius of explosion and the more the number of sparks. At the same time, to ensure the diversity of the population, the Gaussian mutation operator is introduced to generate a certain number of Gaussian mutation sparks.

(4) Choose the best combination into the next generation directly. The other $n-1$ combinations and sparks are selected by certain strategy.

(5) Check whether the current situation satisfies the stopping criterion, if so, the iteration will be terminated,

or the second to fourth steps will be repeated until the stopping criterion is satisfied.

In the combination optimization problem, the fireworks algorithm can finish the iteration after finding the solution that meets the requirements or reaching the set conditions, which is more efficient than the grid search method. Compared with other intelligent optimization algorithms such as genetic algorithm, particle swarm optimization and so on, the unique explosion mechanism and selection strategy of fireworks algorithm can ensure that the population generated by each generation is more diverse, hence it is easier to converge to the global optimum.

C. XGBoost-FWA Short Term Load Forecasting Model

The load data of a power plant in a certain area in the first six days of a week is used as training set, and the load data of 96 moments in the last day is used as test set. Appropriate factors can ensure the accuracy of the model. The external factors selected in this paper are temperature, humidity and rainfall, and the following prediction models are constructed based on these factors:

(1) S-XGBoost-FWA: sequential characteristics;
(2) T-XGBoost-FWA: S-XGBoost-FWA + temperature;
(3) H-XGBoost-FWA: S-XGBoost-FWA + humidity;
(4) R-XGBoost-FWA: S-XGBoost-FWA + rainfall;
(5) M-XGBoost-FWA: S-XGBoost-FWA + temperature + humidity + rainfall.

III. Data Processing and Performance Indicators

A. Interpolation of Missing Data

Data is of the utmost importance for establishing and training models. However, in practical applications, the problem of data missing is unavoidable for various reasons. If an inappropriate interpolation algorithm is adopted, it is equivalent to introducing a lot of noise, which will pollute the data. Some data in this paper is also missing. For different data missing situations, different processing methods are proposed: If more than three consecutive data is missing, discard them, otherwise, too much noise will be introduced into the training data; otherwise, we can use cubic spline interpolation to interpolate missing data.

The principle of cubic spline curve is as follows:

For $n+1$ samples $(x_i, y_i), i = 0,1,\mathrm{L}\ ,n$, where $a = x_0 < x_1 < \mathrm{L}\ < x_n = b$, we have n intervals $[x_i, x_{i+1}], i = 0,1,\mathrm{L}\ ,n-1$. The curve function $S(x)$ is defined in different intervals, and it satisfies the following three conditions:

(1) Interpolation condition. $S(x_i) = y_i$.

(2) Continuous condition. In the whole interval [a, b], the first and second derivatives of $S(x)$ must be continuous.

(3) Cubic polynomial condition. In each interval $[x_i, x_{i+1}], i = 0,1,\mathrm{L}\ ,n-1$, $S(x)$ is a cubic polynomial.

For example, given 11 sample points, the corresponding temperature values at 2, 6 and 7 sample points are missing, and cubic spline interpolation method is used to interpolate. Figure 1 shows the interpolation effect of the missing data. According to the interpolation curve, the missing data value, namely the temperature corresponding to the dotted line, can be obtained.

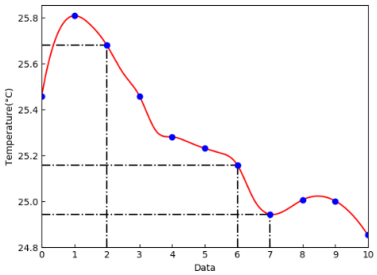

Fig 1. The effect of cubic spline interpolation

B. Data Standardization

There are many factors with different physical properties and dimensions interfering with the load, it is impossible to use these data for analysis directly. In this paper, the normalization method is used to preprocess the original data, so that the range of different factors has the same order of magnitude, and data analysis can be carried out more reasonably. We use a simple and effective min-max standardization method to normalize the data through linear transformation, the normalization function is as follows:

$$x' = \frac{x - \min}{\max - \min} \tag{12}$$

where x is the original data, x' is the result of normalization.

In order to ensure the consistency between the test set and the training set, after normalizing the training data, the test data should also be treated the same way, so that all data can be scaled up and down in proportion. Load data and meteorological data have a wide range of values, the minimum and maximum of a factor in testing set can be very different from that in training set. For individual test data less than minimum or greater than maximum, in order to make the results within the range [0, 1], the following processing is needed:

$$x' = \begin{cases} 0 & x < \min \\ 1 & x > \max \end{cases} \tag{13}$$

C. Prediction Performance Indicators

In this paper, mean absolute error (MAE) is used in the training stage. In the final evaluation of the model performance, root mean square error (RMSE) and mean absolute percentage error (MAPE) are also used.

MAE is the average of absolute error between predicted and actual values. Its calculation is as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - x_i'\right| \tag{14}$$

RMSE is the standard error and the arithmetic square root of MSE, which is the mean square error of predicted and actual values. Its calculation is as follows:

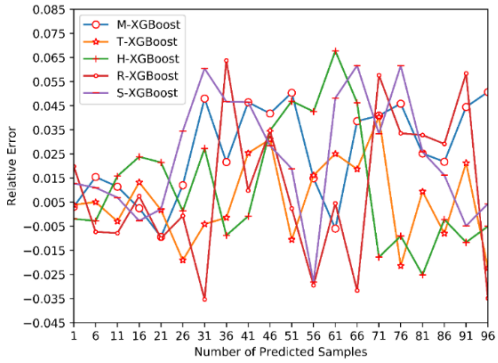

Fig 2. Relative Errors of Predictions of Different XGBoost Models

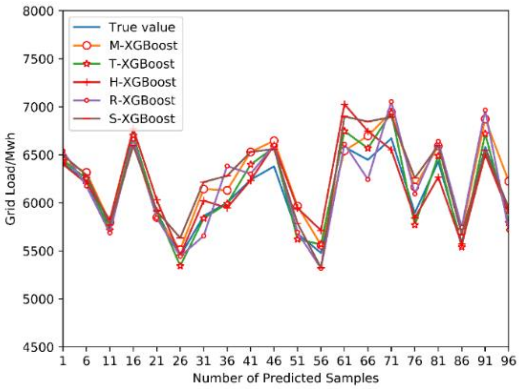

Fig 3. Predicted Load against Actual Load

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_i')^2}{n}} \tag{15}$$

MAPE is the average of relative errors between predicted and actual values:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - x_i'}{x_i}\right| \times 100\% \tag{16}$$

where n is the length of the sequence data, x_i is the actual value of the i-th load data, x_i' is the predicted value of the i-th load data. All three indicators can reflect the overall prediction performance of the model. The smaller the value, the smaller the prediction error, the better the prediction performance of the model.

IV. Experimental Results and Analysis

We select the optimal model of each dimension obtained in optimization and training, and use the data in test set to predict. Figure 2 calculates the relative error of predicted load, Figure 3 shows the predicted load versus actual load. It can be seen that S-XGBooost can predict the future trend of short-term load pretty well, and it is fully qualified without pursuing prediction accuracy. M-XGBoost model with added temperature, humidity, rainfall and other factors has higher prediction accuracy than S-XGBoost model. Meanwhile, we found that T-XGBoost model is better than H-XGBoost and R-XGBoost, indicating that the effect of temperature on load changes is much higher than rainfall and humidity.

Table 1. Comparison of Model Prediction Errors

Model	MAE	MSE	RMSE
M-XGBoost	1.551%	0.041%	1.842%
T-XGBoost	1.024%	0.019%	1.373%
H-XGBoost	1.319%	0.046%	2.108%
R-XGBoost	2.736%	0.127%	3.561%
S-XGBoost	3.010%	0.140%	3.742%

We can see from Table 1 that RMSE of prediction results of T-XGBoost model is less than M-XGBoost model. There is a strong correlation between temperature and humidity. The drastic fluctuation of rainfall itself is likely to cause much higher noise. Hence, even though the prediction results of three models with single factor are better than S-XGBoost with no external factor, due to the redundancy and interference of the data, the prediction results of the model considering all three factors are not as good as the models considering only one factor.

V. Conclusions

This paper combines the real load data of a power plant within a week, selects the most abrupt meteorological factors (temperature, humidity and rainfall) after comprehensive consideration, and builds multi-dimensional prediction model using the feature of multi-core parallel learning of XGBoost. Considering the difficulty in selection of hyper parameters in traditional methods, the fireworks algorithm is introduced into the optimization problem. The experiment proves the feasibility of XGBoost short-term load forecasting under multiple dimensions. But we also find that the T-XGBoost model with only one factor is better than the M-XGBoost model that integrates multiple factors, which indicates more factors are not always better in machine learning, the accuracy of the model will be reduced when there is interference or redundancy among features.

Reference

[1] Fantazzini D, Toktamysova Z. Forecasting German car sales using Google data and multivariate models. International Journal of Production Economics, 2015, 170: 97-135.

[2] Hülsmann M, Borscheid D, Friedrich C M, et al. General Sales Forecast Models for Automobile Markets and their Analysis. Trans. MLDM, 2012, 5(2): 65-86.

[3] HUANG S, WEI Z, DING Q. Bus Load Forecasting Model Based on Stacked Generalization. Proceedings of the CSU-EPSA, 2013,25(03):8-12.

[4] LONG D, LI J, WEI H. Solution of Multi Environment Factor-Influenced Bus Load Forecasting by Rough Set Method. Power System Technology, 2013,37(05):1335-1340.

[5] HAN Y, LI H. SVM Bus Load Forecasting Based on Wavelet Decomposition. Electric Power Automation Equipment, 2012,32 (04):88-91.

[6] LI G, Liu W. Bus Load Short-term Forecast Based on LSSVM and Markov Chain. Power System Protection and Control, 2010,38(11):55-59.

[7] Peng X, HE H, YAO J. Method of Bus Load Forecasting Using BP Neural Network Optimized by PSO. Proc. Of CSU-EPSA, 2010,22(05):146-151.