An Improved AdaBoost Algorithm for Hyperparameter Optimization

Rongfang Gao and Zhanyu Liu*

College of Computer Science and Technology, Xi’an Shiyou University, Xi’an, China
Email: liu_nini@qq.com

Abstract. AdaBoost algorithm is a typical Boosting algorithm, which belongs to a successful representative in the Boosting family. This algorithm can upgrade a weak classifier with a better classification effect than random classification to a strong classifier with high classification accuracy, where n_estimators represents the number of iterations of the base classifier. If the value is too large, it will easily cause the model to overfit, if it is too small, it is easy. The model is under-fitting, and the parameter setting is not set randomly, but according to the current status of the data set. Aiming at the problem that the number of iterations in the AdaBoost algorithm is uncertain, this paper introduces a Bayesian optimization algorithm for hyperparameter tuning, which makes the value of hyper parameter in AdaBoost algorithm suitable for the current data set, and finally obtains a hyperparameter optimization AdaBoost algorithm. The experiment result shows the method that adopt Bayesian optimization algorithm for hyperparameter optimization and apply the optimized hyperparameter value to the AdaBoost algorithm does not only improves the classification accuracy of the AdaBoost algorithm, but also avoids overfitting and underfitting of the model.

1. Introduction

AdaBoost algorithm is a dichotomous iterative algorithm proposed by Yoav Freund and Robert Schapire. There are many shortcomings in model training with AdaBoost algorithm, such as poor setting of the number of iterations, reduction of classification accuracy due to imbalanced data, sensitivity to abnormal samples, time-consuming training and so on. In terms of these shortcomings, many efforts have been done by the researchers. To solve the problem of data imbalance, Viola and Jones proposed an asymmetric AdaBoost method [1]. Li et al. proposed an improved method based on adjusting weight distribution and limiting weight expansion [2]. In terms of the time-consuming training problem, Stan et al. proposed the FloatBoost algorithm, which aims to replace the weak classifier in poor performance, minimize the error rate, and become one strong classifier with effective classification result [3]. Souza et al. adopted the taxonomy to modify the weak classifier in each step through the training process to improve classification accuracy [4].

The grid search algorithm and Bayesian optimization algorithm are commonly used for hyperparameter tuning in the field of machine learning. As hyperparameter tuning is carried out, the grid search algorithm may appear local optimization rather than global optimization, while Bayesian optimization algorithm can realize global optimization. The Bayesian optimization algorithm performs hyperparameter tuning by making use of the information of the previous point that ignored by the test point, assuming a collection function based on the prior distribution, using each new sampling point to test the target function to update the prior distribution of the target function, and then testing the global most likely position point given by the posterior distribution [5]. Therefore, aiming at the problem that
the number of iterations in AdaBoost algorithm is uncertain, this paper proposes an improved method for hyperparameter tuning using Bayesian optimization algorithm, which makes the AdaBoost algorithm hyperparameter value globally optimal during model training.

2. Overview of Related Theories

2.1. AdaBoost Algorithm

Boosting, also known as Boosting Learning or Lifting Method, is an important integrated learning technology, which can enhance a weak classifier with a classification accuracy that is only slightly higher than the one of random guess to a strong classifier with high classification accuracy [6]. AdaBoost algorithm is a typical Boosting algorithm as a successful representative of Boosting. Its basic idea is to train the same training set by a large number of weak classifiers with general classification capabilities, and then to integrate these weak classifiers through combination strategy to get a strong classifier with stronger classification capabilities. According to the basic idea of AdaBoost algorithm, the steps of AdaBoost algorithm solving the classification problem are as follows:

Suppose that here is a given training data set \( T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\} \), wherein \( x \) is the input sample and \( y \) is the category to which the sample belongs.

1. Initialize the weight distribution of the training data. The AdaBoost algorithm assigns a weight to each training data and adjusts the weight in each iteration. First, all samples are assigned the same weight:

\[
D(1) = (\omega_1, \omega_2, \ldots, \omega_m)
\]

Among them: \( \omega_i = \frac{1}{m}, i = 1, 2, \ldots, m \).

2. Train the weak classifier. Train the data based on the current weight, and a weak classifier \( G_k(x) \) is received:

\[
G_k(x): x \rightarrow \{-1, +1\}
\]

Among them, \( k = 1, 2, \ldots, m \) represents the \( k \)-th iteration.

(a) Calculate the classification error rate \( e_k \) of the \( k \)-th \( G_k(x) \):

\[
e_k = P(G_k(x_i) \neq y_i) = \sum_{i=1}^{m} \omega_i I(G_k(x_i) \neq y_i)
\]

(b) Update the weight coefficient \( \alpha_k \) according to the classification error rate \( e_k \) of \( G_k(x) \):

\[
\alpha_k = \frac{1}{2} \log \frac{1-e_k}{e_k}
\]

(4)

\( \alpha_k \) is the importance of \( G_k(x) \) in the final strong classifier. From the above formula, it shows that the larger \( e_k \) is, the smaller the weight coefficient \( \alpha_k \) of \( G_k(x) \) is, and vice versa.

(c) Update the weight distribution \( D(k+1) \) of the training data set according to \( \alpha_k \), and then repeat the next iteration until the number of iteration reaches \( m \) specified in advance:

\[
D(k+1) = (\omega_{k+1,1}, \omega_{k+1,2}, \ldots, \omega_{k+1,m})
\]

(5)
Among them, $\omega_{k,i} = \frac{\omega_0}{Z_k} \exp(-\alpha_i y_i G_k(x_i)) i = 1,2,...,m$, $Z_k$ represents the Normalization Factor:

$$Z_k = \sum_{i=1}^{m} \omega_0 \exp(-\alpha_i y_i G_k(x_i))$$  \hspace{1cm} (6)$$

(3) Integrate the weak classifier through a combination strategy, and get the final strong classifier $G(x)$:

$$G(x) = \text{sign}(f(x)) = \text{sign}(\sum_{i=1}^{m} \alpha_i G_i(x))$$  \hspace{1cm} (7)$$

Among them, $f(x)$ represents a linear combination of weak classifiers. $\alpha_i$ represents the coefficient of the $k$-th iteration. $G_k(x)$ represents the weak classifier of the $k$-th iteration.

$n_{\text{estimators}}$ (number of iterations) is a hyperparameter in the AdaBoost algorithm, and its default value is 50. This value is set before the model training using the AdaBoost algorithm. Different $n_{\text{estimators}}$ values mean that the number of weak classifiers in the AdaBoost algorithm is different, and the weight distribution of the weak classifiers is different, and the degree of fit of the model built using the AdaBoost algorithm is different. Generally speaking, in terms of AdaBoost algorithm for model training, small $n_{\text{estimators}}$ is prone to lead to model underfitting, while large $n_{\text{estimators}}$ is easy to cause overfitting.

2.2. Bayesian Optimization Algorithm

The Bayesian Optimization Algorithm (BOA) is a probabilistic model that uses Bayesian networks as estimated chain dependent relationships of variables, which has significant advantages in solving higher-order optimization problems [7]. Its Bayesian network is a directed acyclic graph with probability labels. Its nodes represent variables in their own discourse domains and the directed arcs represent the relationships between the variables, and the strength of the relationships between the variables is represented by the conditional probability between the nodes and their parent nodes.

The main idea of the Bayesian optimization algorithm for hyperparameter tuning is to update the posterior distribution of the objective function by adding sample points based on a given optimized objective function, and then select the next sample hyperparameter based on the posterior distribution until the experimental posterior distribution basically fits the real distribution.

The Bayesian optimization algorithm keeps the number of parameters constant in performing hyperparameter tuning. Its tuning process is to obtain the posterior probability distribution according to the famous “Bayes theorem” [8]:

$$p(f | D_{\text{tr}}) = \frac{p(D_{\text{tr}} | f) p(f)}{p(D_{\text{tr}})}$$  \hspace{1cm} (8)$$

In above formula, $f$ represents the parameter in the probability model; $D_{\text{tr}} = \{(x_1, y_1), (x_2, y_2), ..., (x_t, y_t)\}$ represents the observed sample set; $x_t$ represents the decision vector; $y_t = f(x_t) + \varepsilon_t$ represents the observed value; $\varepsilon_t$ represents the observation error; $p(D_{\text{tr}} | f)$ represents the likelihood distribution of $y$ under the condition of $f$; $p(f)$ represents the prior probability distribution of $f$, that is, the assumption of the unknown objective function state; $p(D_{\text{tr}})$ represents the marginal likelihood distribution of marginalization $f$. Because of the deposition and integration of the probability density function in the marginal likelihood, it is often difficult to obtain a clear analytical formula. This marginal likelihood $p$ is mainly used in Bayesian optimization for hyperparameter tuning or it is used as a constant; $p(f | D_{\text{tr}})$ represents the posterior probability distribution of $f$, and
the posterior probability distribution describes the confidence level of \( f \) corrected by the observed data \( D_{12} \).

### 3. Improved AdaBoost Algorithm

#### 3.1. AdaBoost Algorithm Improvement Ideas

Aiming at the problem that the number of iterations in the AdaBoost algorithm is uncertain, this paper discusses to use Bayesian optimization algorithm for hyperparameter tuning, and finally get a hyperparameter optimized AdaBoost algorithm. The basic idea of this algorithm is as follows: the random forest algorithm does not only have relatively strong stability but also contains the same n_estimators hyperparameters as AdaBoost. In the defined objective function of hyperparameter tuning, the random forest algorithm is used to build the model and the output result is used as the data object for cross validation. The input to this objective function is the hyperparameters of all random forest algorithms, and the output is the cross-validated AUC mean value. Then, the output result of this function is taken as the data object of the Bayesian optimization algorithm and the model is built with the algorithm. Finally, the maximum n_estimators output from the model is applied to AdaBoost algorithm.

#### 3.2. AdaBoost Algorithm Improvement Design Ideas

- Set \( t = 0 \) and set the initial population \( P(0) \) of the parameter combination.
- Select a candidate solution \( S(t) \) from \( P(t) \).
- Build a Bayesian network \( B \) that meets the requirements under certain selection rules and restrictions.
- Generate a new solution \( O(t) \) according to the joint distribution function of the Bayesian network \( B \).
- Replace the partial solution in \( P(t) \) with \( O(t) \) to form a new population \( P(t+1) \).
- If the termination conditions are not met, back to (2).
- Obtain the optimal solution of the new population \( P(t+1) \).
- Train the final classifier \( G(x) \) of AdaBoost algorithm by the optimal solution of \( P(t+1) \).

### 4. Experiments and Results

#### 4.1. Experiment Preparation

The experiment selects three sets of data sets with different categories, dimensions, and domains with a number of two: Audit, Z-Alizadeh Sani, and Online Shoppers Purchasing Intention. The data sets are from UCI (http://archive.ics.uci.edu/ml/index.php) machine learning database.

Among them, Audit is the 1994 census income data set; Z-Alizadeh Sani is the CAD diagnostic data set in 2017; Online Shoppers Purchasing Intention is the 2018 online purchase intention data set. The experimental data set sample distribution information is shown in Table 1:

| Dataset name                        | Number of data | Dimension | Class |
|-------------------------------------|----------------|-----------|-------|
| Audit                               | 776            | 27        | 2     |
| Z-Alizadeh Sani                     | 303            | 56        | 2     |
| Online Shoppers Purchasing Intention| 12330          | 18        | 2     |

The experiment is run on a personal computer processor Intel(R) Core (TM) i5-5200U CPU @2.20GHZ, memory 4G, 64-bit Windows 10 professional operating system. It is implemented by the Anaconda3 compiler in Python language, and the third-party databases such as pandas, numpy, sklearn and bayes_opt in Python are used.
4.2. Experimental Verification

Three most typical and most commonly used algorithms for classification (SVM, Naive Bayes and AdaBoost) are adopted in the experiment. On three data sets, namely Audit, z-alizadeh Sani and Online Shoppers Purchasing Intention, the corresponding models are trained by using these three algorithms. The accuracy of the model is used to express the classification accuracy of the algorithm (unit: %). The classification accuracy of the three algorithms is shown in Table 2:

| Algorithms      | Classification accuracy |
|-----------------|-------------------------|
|                 | Audit                   | Z-Alizadeh Sani | Online Shoppers Purchasing Intention |
| SVM             | 87.82                   | 62.30          | 84.78                                 |
| Naive Bayes     | 77.56                   | 65.57          | 81.29                                 |
| AdaBoost        | 89.74                   | 72.13          | 84.54                                 |

From Table 2, as we can see, three different data sets are used for training on SVM algorithm, Naive Bayes algorithm, and AdaBoost algorithm to obtain their respective classification accuracy. According to the classification accuracy, it is safe to conclude that AdaBoost algorithm has better classification effect than the other two algorithms.

Characterized by that the Bayesian optimization algorithm can be used for hyperparameters tuning and helping the model to find the optimal hyperparameters that are suitable for the current data set, for AdaBoost algorithm with good classification effect, the Bayesian optimization algorithm is used to optimize the hyperparameters of AdaBoost algorithm. According to the different sizes of the three data sets, the n_estimators hyperparameter ranges of the objective function for hyperparameter tuning by the Bayesian optimization algorithm are set as: (10, 776), (10, 300), (10, 2000). As the improved AdaBoost algorithm for training the model is used, the classification accuracy of the model is significantly improved. The comparison of the n_estimators hyperparameters and classification accuracy between the AdaBoost algorithm and the improved AdaBoost algorithm is shown in Table 3.

| Data set                             | AdaBoost | Optimized AdaBoost |
|--------------------------------------|----------|--------------------|
|                                      | n_estimators | Classification accuracy | n_estimators | Classification accuracy |
| Audit                                | 50        | 89.74              | 505          | 91.67                  |
| Z-Alizadeh Sani                      | 50        | 72.13              | 242          | 73.77                  |
| Online Shoppers Purchasing Intention | 50        | 84.54              | 1996         | 89.62                  |

As can be seen from Table 3, setting the n_estimators hyperparameter range according to the different data set sizes and using the Bayesian optimization algorithm to optimize the parameters, the n_estimators hyperparameter values that suitable for the current data situation are taken as 505, 242, and 1996 respectively, which are the hyperparameter values of AdaBoost algorithm. The experiment result shows that the improved AdaBoost algorithm based on the Bayesian optimization algorithm has higher classification accuracy than the AdaBoost algorithm, which improves by 1.93%, 1.64%, and 5.08% on the three data sets, respectively. Therefore, the method of adopting Bayesian optimization algorithm for hyperparameter tuning and applying the obtained hyperparameter values to AdaBoost algorithm model training can not only improve the classification accuracy but also avoid overfitting and underfitting of the model.
5. Conclusion
Aiming at the problem that the number of iterations in the AdaBoost algorithm is uncertain and n_estimators will cause overfitting and under-fitting problem to the model, this paper introduces a Bayesian optimization algorithm for hyperparameter tuning, which makes the value of hyperparameter in AdaBoost algorithm suitable for the current data set, and finally obtains a hyperparameter optimization AdaBoost algorithm. The experiment result shows that the use of Bayesian optimization algorithm for hyper-parameter tuning improves the classification accuracy of the AdaBoost algorithm and avoids the risk of overfitting and underfitting of the model. However, it is found during the experiment that the running time of the algorithm is relatively long as the amount of data is relatively large, so how to shorten the running time of the algorithm is one of the issues to be studied in the future.

Acknowledgments
This paper is funded by The Graduate Student Innovation and Practice Ability Training Program of Xi’an Shiyou University, Item Number (YCS20153001).

References
[1] Viola P and Jones M 2002 Fast and robust classification using asymmetric AdaBoost and a detector cascade Advances in Neural Information Processing Systems 14 1311-1318.
[2] Li B, Wang Z, Wang W, et al 2004 An improved method of AdaBoost Algorithm Journal of Chinese Computer Systems 25 (5) 869-871.
[3] Li S Z and Zhang Z 2004 FloatBoost learning and statistical face detection IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) 0-1123.
[4] Souza E and Matwin S 2012 Improvements to adaboost dynamic Canadian Conference on Advances in Artificial Intelligence.
[5] Li B and Wang W 2019 Improvement of classification model by NCA dimensionality reduction and Bayesian optimization parameter adjustment Journal of Computer Applications and Software 8 281-287.
[6] Cao Y, Miao Q, Liu J, et al 2013 Research progress and prospect of AdaBoost algorithm Journal of Acta Automatica Sinica 39 (6) 745-758.
[7] Snoek J, Larochelle H and Adams R P 2012 Practical bayesian optimization of machine learning algorithms Advances in Neural Information Processing Systems 4.
[8] Cui J and Yang B 2018 Overview of bayesian optimization methods and applications Journal of Software 29 (10) 176-198.