Research on machine learning framework based on random forest algorithm

Cite as: AIP Conference Proceedings **1820**, 080020 (2017); https://doi.org/10.1063/1.4977376
Published Online: 13 March 2017

Qiong Ren, Hui Cheng, and Hai Han

Research on Machine Learning Framework Based on Random Forest Algorithm

Qiong Ren [a)], Hui Cheng [b)] and Hai Han

School of Mathematics and Computer Science, Jianghan University, Wuhan, China.

[a)]qren@163.com
[b)]huicheng2001@163.com

Abstract. With the continuous development of machine learning, industry and academia have released a lot of machine learning frameworks based on distributed computing platform, and have been widely used. However, the existing framework of machine learning is limited by the limitations of machine learning algorithm itself, such as the choice of parameters and the interference of noises, the high using threshold and so on. This paper introduces the research background of machine learning framework, and combined with the commonly used random forest algorithm in machine learning classification algorithm, puts forward the research objectives and content, proposes an improved adaptive random forest algorithm (referred to as ARF), and on the basis of ARF, designs and implements the machine learning framework.

Key words: Machine learning; Random forest algorithm; ARF.

INTRODUCTION

Machine learning, through the design of some of the computer automatic "learning" algorithms, analyzes the existing data to get the hidden rules, and use these laws to predict and analyze the unknown data. With the rapid development of the mobile Internet era, the generation of massive data and the improvement of industry for calculating speed and cost requirements, the traditional mainframe has been difficult to meet the needs of the industrial sector. As a result, the distributed computing technology comes into being. Distributed computing technology, through the decomposition of a task that can only be solved with huge computing power into many sub tasks, and then assign those sub tasks to many computer nodes for processing, and eventually summarize the calculation results to get the final result.

Random forest algorithm proposed is proposed by Leo Breiman and Adele Cutler, which combines with the "Bootstrap aggregating" method and "random subspace method" method to build a set of decision trees, and through the decision tree set to classify. Random forest contains multiple decision tree classifiers, and the output categories are determined by the mode of decision tree classification results. In the construction of a single decision tree, the random forest algorithm uses two random selection processes: the first is the random selection of training samples, and the second is the random selection of the characteristics attributes of the sample. After all the decision trees are constructed, the final classification result is decided by the method of equal-weight voting.

Advances in Materials, Machinery, Electronics I
AIP Conf. Proc. 1820, 080020-1–080020-7; doi: 10.1063/1.4977376
Published by AIP Publishing. 978-0-7354-1488-4/$30.00

METHOD

Random Forest Algorithm

Random forest is an ensemble classifier, which constructs a group of independent and non-identical decision trees based on the idea of randomization. Random forest can be defined as $\{h(x,\theta_k),k=1,\cdots,L\}$, in which θ_k is a kind of mutual independent random vector parameter, and x is the input data [1]. Each decision tree uses a random vector as a parameter, randomly selects the feature of samples, and randomly selects the subset of the sample data set as the training set.

The construction algorithm of random forest is as follows. k suggests the number of decision tree in the random forest, n indicates the number of sample in training data-set that each decision tree corresponds to, M refers to the feature number of sample, m represents the number of features when carrying out segmentation on a single node of a decision tree, $m<<M$ [2]:

(1) From all the training sample sets, sample for N times with a repeated sampling method, form k group training set (namely bootstrap sampling). Respectively construct decision tree for each training set, samples not being selected form k group data out of bag, referred to as OOB;

(2) For each node of the decision tree, randomly select m features based on this node, and according to the m characteristics, calculate the best segmentation characteristics;

(3) Each decision tree is completely growing without pruning;

(4) Form several decision trees into a random forest model and use the model to identify and classify the unknown data.

Feature Selection of Random Forest Algorithm

Feature selection is the process of using a series of rules to calculate the relative relationship of importance of the characteristics and to rank the characteristics of the data. Feature selection techniques are often used to transform the data in classification analysis, so as to improve the accuracy of classification. In general, feature selection techniques in machine learning are mainly divided into three categories: filter method (Filter), encapsulation method (Wrapper) and integration method.

The following is the introduction to the filtering method. Filtration method is through the statistical method, to give the characteristics with a weight, to carry out feature ranking according to the characteristics of the weight, and then apply some rules to set a threshold, the feature whose weight is greater than the threshold value is retained, otherwise deleted. The feature selection process of the filtering method is operated according to the feature of the data set, which is independent of the specific classification algorithm. There are many common filtering methods, such as Fisher ratio, information gain, and Relief, T-test and variance analysis. In the following, variance analysis will be briefly introduced.

When using variance analysis to test the characteristics of the screening method, calculate the test statistic F value of each feature, and its specific formula is as follows:

Inter group variation:

$$BBS=\sum\nolimits_i n_i\left(\overline{Y}_i-\overline{Y}_{total}\right)^2 \tag{1}$$

Intra group variation [4]:

$$WSS=\sum\nolimits_i\sum\nolimits_j\left(Y_{ij}-\overline{Y}_i\right)^2 \tag{2}$$

n_i is the total of the measured value in the i group, $\overline{Y}_i$ is the average value of the i group, $\overline{Y}_{total}$ is the overall average value, and $\overline{Y}_{ij}$ is the j observed value of the feature in the i group;

Inter group mean square:

$$BWSS = \frac{BSS}{k-1} = \frac{\sum_i n_i \left(\overline{Y}_i - \overline{Y}_{total}\right)^2}{k-1} \tag{3}$$

Intra group mean square:

$$WMSS = \frac{WSS}{N-k} = \frac{\sum_i \sum_j \left(Y_{ij} - \overline{Y}_i\right)^2}{N-k} \tag{4}$$

Where k is the number of groups, N is the total number of observations.
The ratio of the two mean square values is [5]:

$$F = \frac{BMSS}{WMSS} \tag{5}$$

The larger the F value, it indicates that the inter group mean square is greater than intra group mean square. That is to say, the variance between groups is greater than the variation within the group, the differences between the groups far exceeds the total expected value deviation, and there are obvious differences in the average number of each group; on the contrary, the smaller the F value, even close to 0, it suggests that the variance between groups is smaller than the variation within the group, the differences between the groups are very small, and the average value has no obvious differences.

RESULTS

Machine Learning Framework Design

The design objectives of machine learning framework mainly include: for application developers who do not have the knowledge of machine learning, to provide adaptive machine learning algorithm process, to reduce the threshold for the use of machine learning framework; for different using scenarios and requirements, to provide convenient tools and access interface. According to the design goal, the structure view of the machine learning framework is shown in Figure 1 [6].

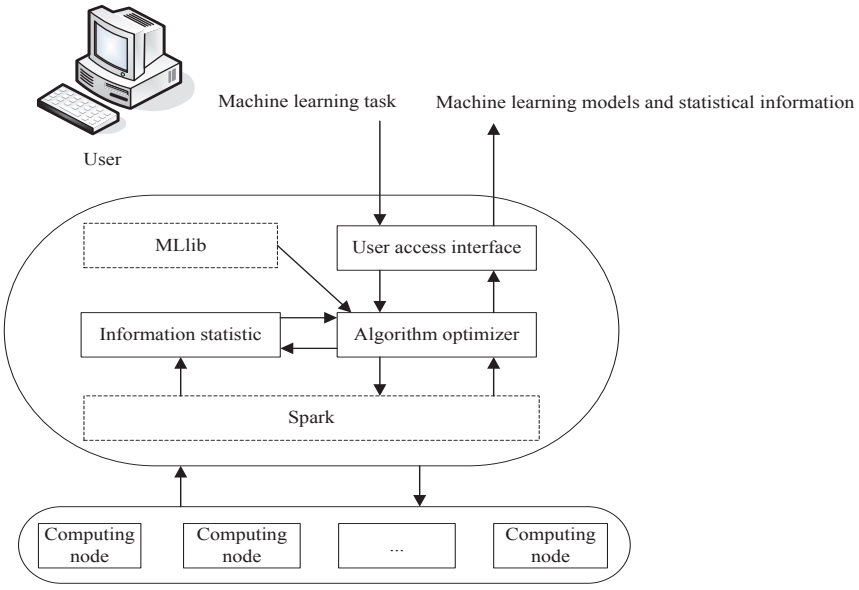

FIGURE 1. Machine learning framework view

Optimal Design and Implementation of Random Forest Algorithm

Optimization Algorithm Design

(1) Algorithm optimization

a) Carry out feature selection and remove noise characteristics

The irrelevant features are filtered in advance: for two-classification problems, use T-test; for multiple-classification problems, use variance analysis; carry out feature selection based on RFE-RF [7], and for the remaining features, according to the importance of the average score, arrange in descending order; use Sequential Forward Selection, referred to as SFS, to select the optimal feature subset.

b) Carry out feature selection and delete redundant features

Use all the features in S to construct a random forest model, denoted by RF_0; using S as the object, only remove one feature in S each time, respectively construct random forest model, denoted as RF_i, in which $0<i<|S|$, then RF_i does not contain the random forest model of the i feature; calculate the data error rate out of pocket and the contribution degree that each feature corresponds to of all the random forest models, the contribution degree of the i feature is the difference value between the error rate RF_i and that of RF_0 [8]; set the threshold value, delete the features whose contribution degree does not exceed the threshold, and the remaining features are the feature subset returned by the final algorithm.

c) Voting strategies for optimizing the random forest algorithm

According to the ten-fold cross validation, segment the original data; the data after segmentation is combined for multiple-group training set and test set, construct random forest model in multiple-group data respectively according to the voting strategy of equal weight and the voting strategy of taking the performance of decision-tree as the weight, and carry out the feature selection and so on steps; accumulate the corresponding errors of the two strategies to compare, and choose the optimal voting strategy; according to the optimal voting strategy, carry out random forest model training in all the data sets.

(2) Process design

Figure 2 is the flow chart of algorithm optimization algorithm (Random Forests Adaptive, referred to as ARF) [9] based on random forest algorithm in machine learning framework

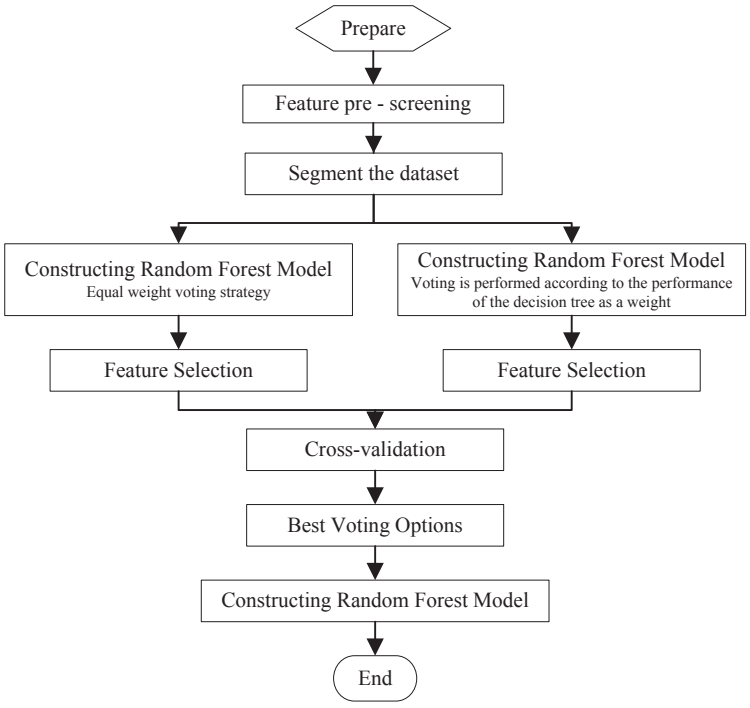

FIGURE 2. Flow chart of ARF algorithm

Optimization Scheme Implementation

The logic framework view of the optimization module of the random forest algorithm is shown in Figure 3 [10], which mainly includes the external interface, the model selection and verification sub-module, the feature pre-filter module and the feature selection sub-module.

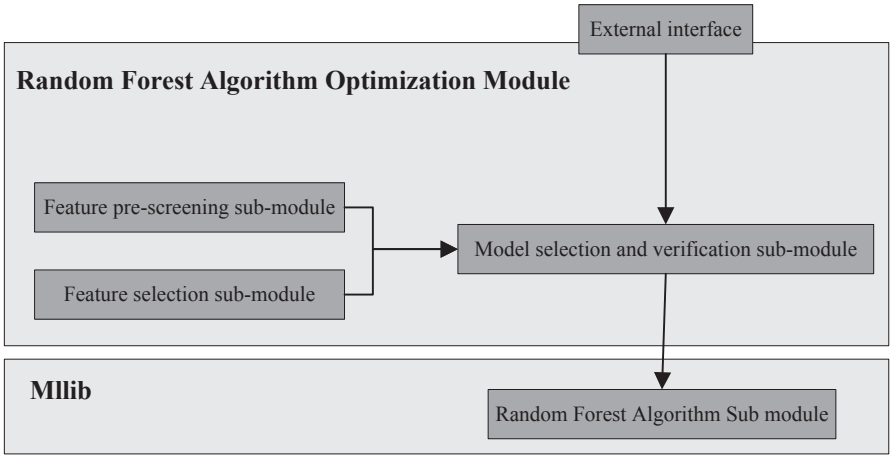

FIGURE 3. Logic architecture view of random forest algorithm optimization module

1) The external interface provides access interface for the user interface layer, not in allusion to the actual user to access, nor for different data sources for adaptation, such as [11]:

def train (input: RDD [Labeled Point]): Random Forests Model

def predict(features: Vector): Double

2) Feature pre-filter module mainly filters the irrelevant features in the loaded data. For two- classification problems, make use of T-test; for multiple-classification problems, make use of variance analysis.

3) Feature selection sub-module is to call the random forest sub-model to repeatedly construct a random forest model, calculate the importance score of feature, and according to the proposed feature selection algorithm to remove noise features and redundant features.

4) Model selection and verification sub-module in allusion to the pre-filtered data, use ten-fold cross validation to select a best random forest optimization scheme, call feature selection sub-module for feature selection, and as for the best optimization scheme, train the model in the global data set, and return the final classification model. The original random forest algorithm is provided by the random forest algorithm sub-module in MLlib [12].

Experimental Verification of Machine Learning Framework

Machine learning framework can not only be used as a static tool, used by calling the API interface, but also be used as a service to deploy, used by calling the RESTful interface. Through the use of API interface, users need to write code and manually manage and utilize the machine learning model; through the use of RESTful interface, users do not need to manage the machine learning model.

In the experiment, the machine learning framework is packed and deployed as a service, and the specific architecture diagram is shown in Figure 4 [13]:

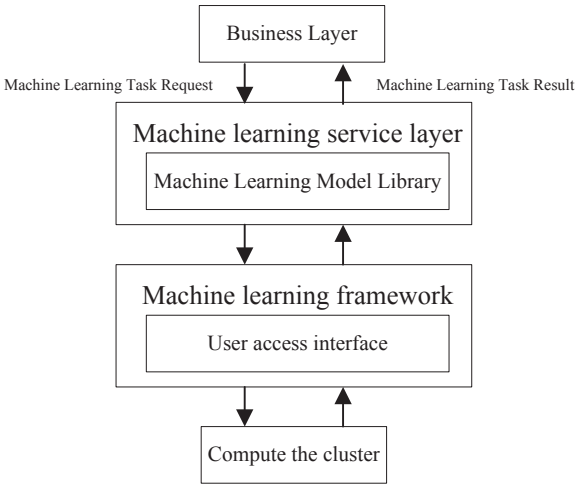

FIGURE 4. Architecture of the laboratory machine learning framework

First of all, the application developers analyze the specific business needs, collect and collate the corresponding data set, and in the business layer, write code to call the RESTful interface that the machine learning service layer provided and submit the machine learning task request;

Secondly, the machine learning service layer parses users access interface and parameter. When the user requests to use the existing learning models in the machine learning model base for data analysis, machine learning service layer gets the model from the machine learning model base to parse, complete machine learning tasks, and return machine learning task request to the user [14]; when the user requests to carry out model learning of data-set, the machine learning service layer transfers the machine learning task request into a specific machine learning algorithm code and package it, and then submit to the machine learning framework to implement;

Finally, the machine learning framework performs machine learning tasks, achieve adaptive data preprocessing, model learning and evaluation model, returns the machine learning model to the machine learning service layer, then the machine learning service layer, based on the request of application developers, uses machine learning model to carry out data prediction and analysis, the machine learning model is saved in machine learning model base, and the corresponding result is returned to the application developers.

The implementation environment of machine learning service layer is shown in Table 1 [15]:

TABLE 1. Implementation environment of machine learning service layer

	Implementation environment
JVM	SUN JRE/JDK 8.0
Web container	Tomcat 7.0
Web front-end	Bootstrap 2.3.2
Web back-end	Spring MVC 2.2.3.RELEASE
Template engine	Velocity 1.7

CONCLUSION

Aiming at the existing limits in random forest algorithm, this paper analyzes and studies the machine learning framework based on random forest algorithm, designs and implements a set of machine learning framework, optimizes the existing limit in random forest algorithm. In addition, it provides a machine learning algorithm interface and specific details for hidden data preprocessing, parameter selection and optimization algorithm for those application developers who do not have machine learning algorithm knowledge to carry out machine learning application development, reducing the threshold for the use of machine learning framework.

ACKNOWLEDGMENTS

Hubei Provincial Department of Education Guidance Project of Scientific Research Program (No. B2016281).

REFERENCES

1. Provost F, Hibert C, Malet J P, et al. Automatic classification of endogenous seismic sources within a landslide body using random forest algorithm [C]//EGU General Assembly Conference Abstracts. 2016, 18: 15705.
2. Bradter U, Kunin W E, Altringham J D, et al. Identifying appropriate spatial scales of predictors in species distribution models with the random forest algorithm [J]. Methods in Ecology and Evolution, 2013, 4(2): 167-174.
3. Hibert C, Provost F, Malet J P, et al. Automated classification of seismic sources in large database using random forest algorithm: First results at Piton de la Fournaise volcano (La Réunion) [C] //EGU General Assembly Conference Abstracts. 2016, 18: 12895.
4. Ahmed O S, Franklin S E, Wulder M A, et al. Characterizing stand-level forest canopy cover and height using landsat time series, samples of airborne LiDAR, and the random forest algorithm [J]. ISPRS Journal of Photogrammetry and Remote Sensing, 2015, 101: 89-101.
5. Schwartz M H, Rozumalski A, Truong W, et al. Predicting the outcome of intramuscular psoas lengthening in children with cerebral palsy using preoperative gait data and the random forest algorithm [J]. Gait & posture, 2013, 37(4): 473-479.
6. Xiao L H, Chen P R, Gou Z P, et al. Prostate cancer prediction using the random forest algorithm that takes into account transrectal ultrasound findings, age, and serum levels of prostate-specific antigen [J]. Asian journal of andrology, 2016.
7. Scerbo M L, Radhakrishnan H, Cotton B, et al. Utilization of the Random Forest Algorithm to Predict Trauma Patient Disposition Based on Pre-hospital Variables [J]. Journal of Surgical Research, 2013, 179(2): 271.
8. Kehoe M, O'Brien K, Grinham A, et al. Random forest algorithm yields accurate quantitative prediction models of benthic light at intertidal sites affected by toxic Lyngbya majuscula blooms [J]. Harmful algae, 2012, 19: 46-52.
9. Li X, Zhai T, Jiao Y, et al. Using Bayesian hierarchical models and random forest algorithm for habitat use studies: a case of nest site selection of the crested ibis at regional scales [R]. PeerJ PrePrints, 2015.
10. Fu B, Wang Y, Campbell A, et al. Comparison of object-based and pixel-based Random Forest algorithm for wetland vegetation mapping using high spatial resolution GF-1 and SAR data [J]. Ecological Indicators, 2017, 73: 105-117.
11. Mascaro J, Asner G P, Knapp D E, et al. A tale of two "forests": Random Forest machine learning aids tropical forest carbon mapping [J]. PloS one, 2014, 9(1): e85993.
12. Bodduluri S, Newell J D, Hoffman E A, et al. Registration-based lung mechanical analysis of chronic obstructive pulmonary disease (COPD) using a supervised machine learning framework [J]. Academic radiology, 2013, 20(5): 527-536.
13. Botu V, Ramprasad R. Adaptive machine learning framework to accelerate ab initio molecular dynamics [J]. International Journal of Quantum Chemistry, 2015, 115(16): 1074-1083.
14. Liu W, Li M, Yi L. Identifying children with autism spectrum disorder based on their face processing abnormality: A machine learning framework [J]. Autism Research, 2016.
15. Li M, Andersen D G, Smola A J. Graph partitioning via parallel submodular approximation to accelerate distributed machine learning [J]. arXiv preprint arXiv:1505.04636, 2015.