An approach to determining the search range boundaries of optimal parameters values for the random forest algorithm

L A Demidova¹ and M S Ivkina²

¹ MIREA – Russian Technological University, 78, Vernadsky avenue, 119454 Moscow, Russian Federation
² Regional State Autonomous Institution of Continuing Education “Digital Education Center”, 80, Dzerzhinsky street, Ryazan, Russian Federation

E-mail: maria.ivkina996@gmail.com

Abstract. The problem of determining the search ranges for the optimal values for the main parameters in the random forest (RF) algorithm in order to reduce the time spent on developing RF classifier has been considered. The aim of the work is to obtain formulas for determining the search range for the values of RF classifier parameters. Formulas are obtained based on the results of experimental research on the development of RF classifiers using various sets from machine learning data repositories. The results of experimental research on the development of RF classifiers using training and test sets formed on the basis of the analyzed datasets have been presented. Formulas for graphical dependencies for assessing the quality of classification on the test set and development time have been obtained in general form. The recommendations on the application of the proposed formulas in the development of RF classifiers have been given.

1. Introduction

A large number of machine learning algorithms, which are successfully used to solve various data classification problems, is known nowadays [1 – 7]. However, those algorithms require “fine” tuning of the values of their parameters to ensure high quality classification solutions.

It is obvious that enumeration of values of parameters of the classification algorithm on the grid or their search using modern evolutionary optimization algorithms can be accompanied by significant time costs, and use of values of parameters of algorithms “by default” does not guarantee that the resulting classifier will be characterized by high values of the classification quality indicators. Therefore, it will be necessary to use algorithms for finding the optimal values of the classifier parameters. A herewith, it is advisable to set the search range for each parameter. It is obvious that it is necessary to develop formulas for determining the search ranges for the values of the classification algorithm parameters, since this will minimize the time spent on developing the desired classifier.

One of currently actively used machine learning algorithms is random forest (RF) algorithm [1, 2]: it can be used both in development of regression models and in development of data classifiers.

It should be noted that a random forest can be used as an independent classifier, as well as part of hybrid classifiers. In the case of developing a hybrid classifier, an RF algorithm can be applied, for example, as part of the SVM-RF classifier [3] to clarify the class belonging of objects near the class boundary, found using the SVM classifier [5, 6].
When working with the algorithm of a random forest, it is necessary to search for the optimal values of its parameters. In the simplest case, this can be done by iterating over the grid, which will be accompanied by significant time costs. In addition, one can apply any evolutionary optimization algorithm, for example, the genetic algorithm [8], the particle swarm algorithm [9], etc., by coding the solution with the optimized parameters. But even in this case time costs will be large. Obviously, it is necessary to try to reasonably narrow the range of searching for the optimal values of the RF algorithm parameters.

In the context of solving the problem of developing an RF classifier, it is proposed to carry out experimental studies, the results of which can be used to form recommendations for determining the boundaries of the search ranges for the optimal values of the main parameters of the algorithm on the basis of the studies.

In particular, to determine the boundaries of the search range, it was decided to perform the experiment, during which we studied for 40 datasets the dependences of the quality of the classifier and the development time of the random forest on the number of trees, the number of features when dividing a node and the depth of the tree.

2. Theoretical part

2.1. Decision tree

A decision tree is based on logical diagrams which make it possible to obtain a decision on the classification of an object using answers to a hierarchically organized system of questions. The tree is composed of “leaves” and “branches”. The edges (“branches”) of the decision tree contain the attributes on which the objective function depends, the “leaves” contain the values of the target label, and the remaining nodes contain questions asked at the current hierarchical level and depending on the answers received in the previous nodes [1, 2, 4].

It is necessary to train the classifier before classifying data. Training is performed on a training set randomly extracted from the analyzed dataset, and includes the search for optimal threshold values of parameters or binary partitions for features, based, for example, on the requirement to maximize the entropy index of heterogeneity in subsets.

The construction of the tree begins from the root, in which, taking into account the values of the inhomogeneity index for each feature of the objects, the optimal feature is found, with the help of which the set is divided into subsets. Further, each of the subsets is divided into other subsets based on its optimal feature. The split continues until the current subset meets the stopping criterion and is declared a sheet with a class label. As a stopping criterion, the criterion for achieving complete homogeneity in one of the classes can be used. The tree can always be built if the training set does not contain objects with the same value of each of the features belonging to different classes.

The following can be used as a criterion for splitting a tree top when solving a classification problem:

The following can be used as a criterion for splitting a tree top when solving a classification problem:

- entropy index of heterogeneity:

  \[ \gamma_e = -\sum_{i=1}^{L} P_i \cdot \ln(P_i) , \]

- Gini index:

  \[ \gamma_g = 1 - \sum_{i=1}^{L} P_i^2 , \]
where $p_i$ is the proportion of objects of the $l$-th class (that is, the proportion of objects with a label $\eta_l$) in the subset associated with the top of the tree; $l = \overline{1, L}$; $p_i = N_i / N$; $N_i$ is the number of objects of the $l$-th class in the subset; $L$ is the number of classes; $N$ is the number of objects in the subset.

A herewith, the entropy index of heterogeneity should be maximized, and the Gini index should be minimized.

### 2.2. Random Forest

The random forest is an ensemble decision tree algorithm proposed by Leo Breiman and Adele Cutler. This algorithm is one of the most popular and efficient algorithms for solving classification and regression problems. The random forest is based on bagging, which implements such a method of constructing compositions of classifiers, in which classifiers are trained independently of each other [1, 3].

Each tree is constructed from a subset obtained from the original training set using a bootstrap (return sets). When constructing each tree at the stages of nodes branching, only a fixed number of randomly selected features of the training set are used and a complete tree is built, that is, each leaf of the tree contains observations of only one class. In classification problems based on a random forest, the decision is made by a majority vote.

The random forest algorithm can be represented by the following sequence of steps:

1. Formation of a bootstrap subset from the initial training set for each tree.
2. Construction of the obtained subset of a non-truncated decision tree with a minimum number of observations at the nodes using the following steps:
   - randomly select a certain number of features from the initial set of features, while it is recommended to take the root of the total number of features;
   - select the feature that provides the best splitting from the selected number of features;
   - divide the selection corresponding to the processed node into two subsets.
3. Formation of an ensemble of decision trees. Classification of test data by voting: the class of some object is the class which was chosen by the greater number of trees.

The main parameters of RF classifier are the following [1, 3]:
- number of trees;
- the number of features by which the splitting is realized in the node of the tree;
- maximum depth of trees.

### 3. Description of experiments

In order to determine how strongly the values of the parameters of RF classifier affect the quality of the classification, it is necessary to train many classifiers with different values of the parameters on different datasets. During the experiments, 40 different datasets from data repositories for machine learning were considered [10, 11]. Moreover, each dataset was randomly divided into training and test sets in a ratio of 80:20.

Since the quality of the developed classifiers was assessed not only by Accuracy indicator, which makes it possible to estimate the proportion of objects for which the classifier made the right decision, but also by the development time, it was decided that it would be expedient to develop RF classifiers on different computers. In the course of the experiments, computers with the following characteristics were used:
- PC 1 (processor: Intel (R) Core (TM) i3-2100 CPU 3.10GHz 3.10GHz, 2 cores; RAM: 8 GB; 64-bit operating system);
- PC 2 (processor: Intel(R) Core(TM) i5-3337U CPU 1.80GHz 1.80GHz, 2 cores; RAM: 6 GB; 64-bit operating system);
- PC 3 (processor: Intel(R) Core(TM) i5-3330 CPU 3.00GHz 3.20GHz, 2 cores; RAM: 4 GB; 64-bit operating system).
RF classifiers were trained in Python 3, which has an implementation of the classification algorithm based on a forest of random trees, which allows to tune the above parameters of RF classifier [1, 3].

The following ranges were defined for the values of RF-classifier parameters:

- for the number of trees $r$:
  \[ 10 \leq r \leq 200; \]

- for the number of features $Features$:
  \[ 1 \leq Features, \]
  where $maxFeatures$ is the total number of features;

- for maximum tree depth $depthTree$:
  \[ 1 \leq depthTree \leq maxFeatures. \]

4. Experimental results

As a result of the experiments, graphs of the dependences of the quality and development time of the classifier on various parameters of RF classifier were obtained. In figures 1–6, the graphs obtained during the use of the Bank-additional dataset [12] are shown. This dataset can be used to train the binary classifier to determine whether the client will sign the urgent deposit. The object of this dataset is described by 20 features. The total number of objects is 41188 (a herewith, the training set will contain 32951 objects, the test set will contain 8237 objects).

4.1. Number of features when dividing a node

The dependence of the classification quality on the number of features when dividing a node has the shape of a parabola (figure 1).

The dependence of the development time on the number of features has a linear form (figure 2). Therefore, we should not rely on the time of the classifier development when identifying formulas for the search range boundaries.

![Figure 1. The dependence of the classification quality on the number of features.](image-url)
Figure 2. The dependence of the development time on the number of features.

To derive the calculation formulas, the probability of choosing the best separating feature in the node was used (table 1). Based on the data in table 1, we can say that the formula for calculating the lower and upper boundaries is the following:

\[ P_{\text{features max}} \times P_{\text{Features max}} \],

where \( P \) is the probability of a better feature.

We suggest to use \( P=0.3 \) for the lower boundary and \( P=0.8 \) for the upper boundary.

Table 1. The probability of choosing the best separating feature in the nodes.

| node\feature | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1            | 0.100 | 0.200 | 0.300 | 0.400 | 0.500 | 0.600 | 0.700 | 0.800 | 0.900 | 1     |
| 2            | 0.010 | 0.040 | 0.090 | 0.160 | 0.250 | 0.360 | 0.490 | 0.640 | 0.810 | 1     |
| 3            | 0.001 | 0.008 | 0.027 | 0.064 | 0.125 | 0.216 | 0.343 | 0.512 | 0.729 | 1     |
| 4            | 0     | 0.002 | 0.008 | 0.026 | 0.063 | 0.130 | 0.240 | 0.410 | 0.656 | 1     |
| 5            | 0     | 0     | 0.002 | 0.010 | 0.031 | 0.078 | 0.168 | 0.328 | 0.590 | 1     |
| 6            | 0     | 0     | 0.001 | 0.004 | 0.016 | 0.047 | 0.118 | 0.262 | 0.531 | 1     |
| 7            | 0     | 0     | 0     | 0.002 | 0.008 | 0.028 | 0.082 | 0.210 | 0.478 | 1     |
| 8            | 0     | 0     | 0     | 0.001 | 0.004 | 0.017 | 0.058 | 0.168 | 0.430 | 1     |
| 9            | 0     | 0     | 0     | 0     | 0.002 | 0.010 | 0.040 | 0.134 | 0.387 | 1     |
| 10           | 0     | 0     | 0     | 0     | 0.001 | 0.006 | 0.028 | 0.107 | 0.349 | 1     |
| 11           | 0     | 0     | 0     | 0     | 0     | 0.004 | 0.020 | 0.086 | 0.314 | 1     |
| 12           | 0     | 0     | 0     | 0     | 0     | 0.002 | 0.014 | 0.069 | 0.282 | 1     |
| 13           | 0     | 0     | 0     | 0     | 0     | 0.001 | 0.010 | 0.055 | 0.254 | 1     |
| 14           | 0     | 0     | 0     | 0     | 0     | 0.001 | 0.007 | 0.044 | 0.229 | 1     |
| 15           | 0     | 0     | 0     | 0     | 0     | 0     | 0.005 | 0.035 | 0.206 | 1     |
The brown left vertical lines in figures 1 and 2 show the lower bound for the number of features when dividing a node. The brown right vertical lines in figures 1 and 2 show the upper boundary for the number of features when dividing a node.

4.2. Depth of tree
The dependence of the classification quality on the tree depth has a logarithmic form. To obtain the formula for the upper boundary, we use the graph of the dependence of the classification quality on the tree depth (figure 3). In the general case, the classification quality value becomes close to the maximum at such a value of the tree depth, at which the set is most quickly divided into classes.

**Figure 3.** The dependence of the classification quality on the tree depth.

**Figure 4.** The dependence of the development time on the tree depth.
To obtain a formula for the lower boundary, we used the dependence of development time from the tree depth (figure 4). Figure 4 shows that there comes a moment when the time for developing the classifier does not increase. This means that the tree branching has stopped, since the condition of the minimum number of objects has been reached, that is, there is one object in each leaf, and there is no sense in developing a model with a greater tree depth.

Thus, the formula for calculating the lower boundary is the following:

$$\log_2(\text{countClass})$$,
where \( \text{countClass} \) is the number of classes in the dataset, and the formula for calculating the upper limit is is the following:

\[
\frac{2}{\text{Features}\text{max}}
\]

where \( \text{Features}\text{max} \) is the number of features in the dataset.

The brown left vertical lines in figures 3 and 4 show the lower bound for the tree depth. The brown right vertical lines in figures 3 and 4 show the upper boundary for the tree depth.

4.3. The number of trees
The dependence of the classification quality on the number of trees has a logarithmic form (figure 5). The dependence of the development time on the number of trees has a linear form (figure 6). Therefore, we should not rely on the time of the classifier development when identifying formulas for the search range boundaries.

To determine the formulas, it was assumed that the range limits depend on the number of objects in the dataset [8], that is, the larger the dataset, the greater the values of the lower and upper range boundaries. To support this assumption, values for the lower and upper boundaries were subjectively determined for each dataset.

The lower boundary will be the value of the number of trees, at which the value of the classification quality indicator \( \text{Accuracy} \) stops sharply increasing.

The upper boundary will be the value of the number of trees at which the value of the classification quality indicator \( \text{Accuracy} \) stops changing and fluctuates around its current maximum value.

The brown left vertical lines in figures 5 and 6 show the lower bound for the number of trees. The brown right vertical lines in figures 5 and 6 show the upper boundary for the number of trees.

The obtained values of the lower and upper boundaries for the number of trees will be displayed on the graph, having previously sorted the datasets by increasing the number of objects in them.

To derive the formulas, the lower envelope for the lower boundary and the upper envelope for the upper boundary were built.

![Figure 7](image_url)

**Figure 7.** The boundaries of the search ranges.
As a result, it was possible to determine the green and yellow approximation lines of the boundaries so that all previously identified values of the range boundaries were between them (figure 7).

The least squares method was used to build the approximation lines. This made it possible not only to determine the most accurate dependence for the values of the boundaries of the search ranges on the number of objects in the dataset, but also to build the functions of the obtained graphs.

For a more complete coverage of the values of the range boundaries, it was decided to increase the area between the curves of approximation. As a result, the following formulas were obtained for calculating the boundaries.

To calculate the value of the number of trees along the lower boundary, it is proposed to use the formula:

$$7 \cdot \ln(n_{\text{obj}}) - 40,$$

where $n_{\text{obj}}$ is the number of objects in the dataset.

To calculate the value of the number of trees along the upper boundary, it is proposed to use the formula:

$$8 \cdot \ln(n_{\text{obj}}) + 45.$$

4.4. Comparative analysis results

To assess the effectiveness of the obtained formulas in time, the experiments were carried out on PC 3. A herewith, the optimal values of the parameters were selected on the Bank-additional dataset using the grid search method with the step of 1 for each parameter with the assessment of the computation time (table 2). In the first experiment, the values of the range boundaries for each of the parameters were obtained using the proposed formulas. In the second experiment, the boundaries of the ranges of parameter values were chosen so that these ranges with the very high probability contain the optimal values of the parameters.

| Table 2. Total time to find the optimal parameter values. |
|-----------------------------------------------|
| Number of experiment | Classifier parameter | Initial value | Final value | Computation time |
|----------------------|----------------------|---------------|-------------|------------------|
| 1                    | Number of trees      | 22            | 116         | 53 min.         |
|                      | Number of features   | 6             | 16          |                  |
|                      | Depth of tree        | 4             | 10          |                  |
| 2                    | Number of trees      | 10            | 200         | 16 h. 40 min.   |
|                      | Number of features   | 1             | 20          |                  |
|                      | Depth of tree        | 1             | 20          |                  |

As can be seen from table 2, the use of the proposed formulas has significantly reduced the time for search the optimal values of the classifier parameters.

5. Conclusion

The proposed study was carried out in order to ensure the minimization of time spent in the development of RF classifier for the selection of the optimal values of such parameters as the number of trees, the number of features when dividing a node, the tree depth. The obtained results allow to conclude that it is advisable to use the proposed approach to determining the search ranges boundaries for the optimal values of RF classifier parameters in order to reduce the time spent on its development.
We came to the conclusion that the proposed approach to determining the boundaries of the ranges of the RF classifier parameters can be used both in the development of independent RF classifiers, as well as in the development of hybrid classifiers, for example, in the development of the SVM-RF classifier [3] to find out using the RF classifier belonging to a class of objects near the class boundary, found using the SVM classifier [5, 6].

The goal of further research is to develop a similar approach to finding the boundaries of the search ranges for optimal parameter values for Isolation Forest algorithm [13], which is used, in particular, to develop classifiers based on imbalanced datasets and to identify outliers in datasets.

References

[1] Breiman L 2001 *Machine Learning* 45 5 – 32
[2] Lior Rokach and Oded Maimon 2008 Data Mining with Decision Trees: Theory and Applications (New York: World Scientific Publishing Co.)
[3] Demidova L, Klyueva I and Pylkin A 2019 *Procedia Computer Science* 150, 455 – 461
[4] Chesalin A N, Grodzenskiy S Ya, Nilov M Yu, Agafonov A N 2019 Rossiiskii tekhnologicheskii zhurnal (Russian Technological Journal) 7(5) 20–29 (in rus).
[5] Vapnik V N 1998 *Statistical Learning Theory* (New York: Wiley)
[6] Schölkopf B and Smola A J 2002 Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning) (Cambridge: MIT press)
[7] Schmidhuber J 2015 *Neural Networks* 61 85 – 117
[8] McCall John 2005 *Journal of Computational and Applied Mathematics* 184(1) 205 – 222
[9] Zhang Yudong, Wang Shuihua and Ji Genlin 2015 *Mathematical Problems in Engineering* 2015
[10] Demidova L and Ivkina M S 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency (SUMMA) 518 – 522
[11] Demidova L and Ivkina M 2020 9th Mediterranean Conference on Embedded Computing 1 – 4
[12] Machine Learning Repository: bank-additional URL: http://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[13] Liu F T, Ting K M and Zhou Z 2008 *Eighth IEEE International Conference on Data Mining* 413–422