Implementation and Evaluation of a Random Forest Machine Learning Algorithm

Viachaslau Sazonau sazonauv@cs.manchester.ac.uk

University of Manchester, Oxford Road, Manchester, M13 9PL,UK

Abstract

This work is aimed to evaluate Random Forest (RF henceforth) classification performance and explore its properties. Implementation and all experiments are accomplished in Matlab environment using datasets *heart* and *breast*. Evaluation of RF against its two main parameters is completed. RF classification performance is compared with a single tree classifier using ROC curves. Variable importance is estimated for both datasets using RF. Finally, variable selection using importance ranks influence on RF classification rates is investigated.

1. Introduction

Nowadays, a machine learning algorithm called Random Forest (RF) is widely considered to be a one of most accurate classifiers that attracts attention of many researchers in the area. This work is aimed to investigate its properties, capture behavior on two datasets and evaluate the algorithm classification performance.

The RF algorithm has become an intense research area after its original proposal (Breiman, 2001). The algorithm is essentially based on decision trees working in an ensemble. It is developed from two successful approaches suggested earlier.

The first one is an ensemble of trees where each tree is growing while training on a sample obtained from the training set via bagging *without* replacement. This is a known technique from ensemble learning methodology where generalization error is decreased due to combining decisions (or so-called votes) of multiple learners which are usually weak and unstable individually. The second approach is random split selection for a decision tree. This split is chosen randomly from a subset of best splits.

Thus, these two ideas led finally to the basis for RF algorithm. It generally applies two mechanisms: building an ensemble of trees via bagging *with* replacement (bootstrap) and a random selection of features at each tree node. The first one means that any example selected from the training set can be selected again. Each tree is grown using the obtained bootstrap sample. The second mechanism performs random selecting a small fraction of features and further splitting using the best feature from this set. The size of a fraction (i.e. the number of features to select) is fixed within the algorithm execution.

This article is organized in 5 sections. Basic RF algorithm and its common properties are described in Background section. In addition, this section contains formal techniques used during investigation and outlines its scope revealing the goals. All experimental work is captured in Experiments section where statistical tests are described and main results are presented in graphs. The fourth section analyses experimental results and the fifth section makes conclusions.

2. Background

Despite the success of previous approaches in improvement of classification accuracy, each one has its own drawbacks. The technique of combining multiple learners in an ensemble shows the best performance (i.e. the smallest generalization error) when the basic learners are least correlated. In other words, generalization error rises as the correlation between learners in the ensemble increases. Therefore, the ensemble should be gathered from weak unstable learners since strong learners are likely to be more correlated and tend to overfit which may outweigh improvement of individual classification strength.

RF algorithm solves described challenge by minimizing correlation while maintaining strength. It is achieved by injecting the randomness into the training process. Particularly, random selection of features results in diverse learners which are still individually strong due to splitting using the best feature from the random fraction. Another property increasing diversity is that trees are not pruned during growing. Instead, growing is stopped when a leaf size limit is reached.

In fact, RF algorithm has only two main parameters affecting its performance considerably. The number of trees *m* in an ensemble to grow and the number of features *k* to select randomly at each split. Moreover, it is naturally suitable for multiclass classification challenges because it consists of tree classifiers. Unlike many classifiers cause growing number of parameters to tune while combining them as multiple learners in an ensemble in order to deal with multiclass classification, RF algorithm retains only two mentioned parameters mainly to consider. The RF algorithm is presented below.

Algorithm 1 Random Forest

Input: dataset $T = (\mathbf{x}, y)$, number of trees *m*, number of random features *k*

Output: *RF*, a set of grown trees

Initialize *RF*

for *i* = 1 **to** *m* **do**

T' ← *bootstrap(T)*

Tree ← *trainDT(T', k)*

add *Tree* to *RF*

end for

Algorithm 1 shows the technique of building an ensemble of decision trees using bagging. The function *trainDT(T',k)* performs training of a decision tree on a bootstrap sample *T'* selecting *k* features randomly at each split. The process of training a decision tree needs clarification and described in Algorithm 2.

Algorithm 2 Decision Tree

Input: a sample $T = (\mathbf{x}, y)$, number of random features *k*, a leaf size limit *lsize*

Output: *Tree*, a trained decision tree

Initialize *Tree, fnum* as a total number of features, *tnum* as a predefined number of thresholds

function *trainDT(T, k)*

if *sizeof(T)* <= *lsize* **then**

label ← *indexof(max* in *histogram(T))*

else

fraction ← *random(k* from *fnum)*

thresholds ← *random(tnum)*

(f, t) ← *max* of split function in *fraction* and *thresholds*

(leftT, rightT) ← *split(T, f , t)*

add left child ← *trainDT(leftT, k)*

add right child ← *trainDT(rightT, k)*

end if

end function

There are several differences with common ID3 algorithm. The first one is termination criteria: tree growing (or adding new child nodes) is stopped when a minimal node size is reached, that is the node becomes a leaf and the most common class label is assigned. No pruning is carried out that increases ensemble diversity and decreases correlation.

Another property is selection of a random fraction of features and choice of the best feature in the fraction to perform the next split. A set of *tnum* random thresholds is generated for each feature from the fraction. The size of fraction is recommended to be small: $k = \sqrt{fnum}$ or $k = \log(fnum)$ are commonly used values (Breiman, 2001). A pair of values feature/threshold which maximizes the split function is fixed and used for splitting. The general form of impurity based split functions (Robnik-Sikonja, 2004) is:

$$I(\nu_j) = imp(y) - \sum_j \frac{|T_{\nu_j}|}{|T|} imp(y \mid \nu_j) \qquad (1)$$

where ν_j defines the set of split indexes, $imp(y)$ - impurity of class labels before the split, $imp(y \mid \nu_j)$ - impurity of class labels after the split on ν_j. Higher values of *I* are likely to cause better splits. The most commonly applied split functions are Gini impurity measure:

$$I_G = \sum_i p_i(1-p_i) - \frac{|T_l|}{|T|}\sum_i p_i^l(1-p_i^l) - \frac{|T_r|}{|T|}\sum_i p_i^r(1-p_i^r) \qquad (2)$$

and information gain (entropy reduction):

$$I_E = H(T) - \frac{|T_l|}{|T|}H(T \mid T_l) - \frac{|T_r|}{|T|}H(T \mid T_r) \qquad (3)$$

where $H(T) = -\sum_i p_i \log p_i$ - the entropy function, T_l, T_r - left and right splits correspondingly, p_i, p_i^l, p_i^r - probabilities of class *i* in samples T, T_l, T_r.

Thus, randomness is introduced into the learning process of each tree and completed ensemble consists of diverse learners. One of valuable properties of RF is that it does not overfit as more trees are added. Instead, the generalization error declines (Breiman, 2001). The upper bound for the generalization error is given by:

$$GE \le \bar{\rho}\left(\frac{1}{s^2} - 1\right) \qquad (4)$$

where $\bar{\rho}$ - the mean value of correlation, s – strength of the ensemble. Hence, in order to lower the error bound considerably, correlation should be decreased while strength increased.

Once RF is trained, it is treated as a completed ensemble of base learners which are trees. In order to perform classification of arbitrary example $\boldsymbol{x}$, each learner produces a decision individually and then decisions of all learners in the ensemble are combined to generate a decision of the ensemble as a whole. The easiest way to perform that is simple voting where all learners are equal and the final decision is derived as a majority vote. A more general approach is to assign weights to each learner making some of them more influential than others. The class prediction becomes a weighted sum of individual ones (Alpaydin, 2010):

$$\tilde{y}_i = \sum_j w_j d_{ij} \tag{5}$$

where $w_j \geq 0, \sum_j w_j = 1$ - weights which may depend on the training error of a corresponding learner, $\tilde{y}_i$ - the prediction for class i. The final decision is given as:

$$\tilde{y} = \arg\max_i \tilde{y}_i \tag{6}$$

If weights are considered as approximations of prior probabilities for learners and decisions as conditional likelihoods, Bayesian combination scheme can be derived from (5):

$$P(C_i \mid x) = \sum_{j=1}^{m} P(M_j) P(C_i \mid x, M_j) \tag{7}$$

where $M_j, j = 1...m$ - base learner models. All mentioned combination schemes are potentially applicable to RF as it is an ensemble of trees.

There is one more property which allows applying RF not only for classification but for feature interpretation and selection as well. This is consequently derived from the fact that trees actually select the best feature from a random fraction at every split. Therefore, tree rules are built on the most important features while training. Hence, corruption of less important features does not lead to significant variance of tree error rate while affection of most important ones actually does. This idea can be used to formulate a technique for assessment of feature (or variable) importance.

An out-of-bag sample (*OOB* henceforth) is considered for each tree in RF and associated with its corresponding tree. OOB_j contains examples which are not used to train tree t (Genuer et al, 2010). The test error on this sample for tree t is denoted as $err(OOB_j)$. A new sample OOB_j^f is obtained from OOB_j by permuting the values of feature f randomly. Then evaluation of tree t on sample OOB_j^f is performed and the test error is captured. Finally, importance of feature f is estimated as average growth of error rate across all m trees in RF:

$$VI(f) = \frac{1}{m} \sum_{j=1}^{m} \left[err\left(OOB_j^f\right) - err\left(OOB_j\right) \right] \tag{8}$$

ROC curves are used intensively in Experiments section in this article in order to evaluate and visualize classifiers performance. This makes it necessary to describe briefly this technique here.

Assessing and comparing classifiers only by its average error rates and deviations is not enough for many practical applications, particularly where unequal error costs take place. ROC curves are a valuable and helpful tool to deal with these cases. For every dataset it is possible to calculate a confusion matrix which consists of four rates: true positive, false positive, false negative, and true negative. A basic ROC graph shows the true positive rate against the false positive rate, hence, every binary classifier which is described by its confusion matrix is displayed as a single point on the graph. The point (0, 1) represents perfect classification in ROC space. The points laying northwest to others on the graph are considered as better ones. The diagonal line connecting points (0, 0) and (1, 1) reflects performance of a randomly guessing classifier.

Of course, a single ROC point is poor representation of classifier's behavior. There are several approaches to produce a whole curve showing classification performance in ROC space (Fawcet, 2006). A proposed approach for efficient generation of ROC curves is applied in this work. Threshold averaging technique (Fawcet, 2006) is performed to get final curves with vertical and horizontal deviations.

3. Experiments

All experimental tests investigate two datasets. The first one is *heart* which contains 270 examples with 13 features. The second one is *breast* consisting of 569 examples with 30 features. The examples in each dataset are marked with labels. Both datasets include examples of two classes only.

Recommended parameter values for RF, i.e. the number of features in a fraction $k = \sqrt{fnum}$, leaf size limit *lsize* = *5,* the number of random thresholds *tnum* = *5*, are used throughout all experiments unless explicitly stated otherwise. Information gain measure (3) is applied as splitting criteria. An experimental test has shown there is no significant difference in performance while comparing to Gini impurity on both datasets. It is also revealed that RF with weighted and Bayesian combination scheme performs just slightly better on the given datasets than RF with simple voting. Hence, the choice of the combination

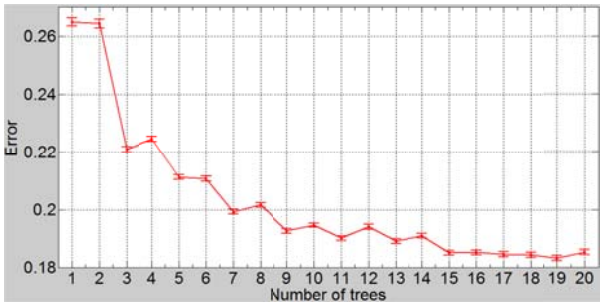
scheme does not affect classification performance results considerably.

Considering a small size of explored datasets, two-fold cross validation is applied in all experiments. More specifically, the dataset is splitted to two equal parts. Examples for each fold are selected randomly from the original dataset. After that, the first RF classifier is trained on the first fold and tested on the second one. In opposite, the second RF classifier is learned on the second fold and tested on the first one. This procedure ensures that each of two classifiers does not experience examples which another one does. Both classifiers are also become trained and tested on the samples of equal size. Test results (whether it is a test error or a ROC curve) are captured for each classifier. A mean test error is calculated from test errors of two classifiers. The whole procedure is repeated multiple times and folds are formed via random selection of examples on each step. Finally, the results of each step are processed and average values with confidence intervals (95%) are plotted on a graph. Described technique is used in all experiments.

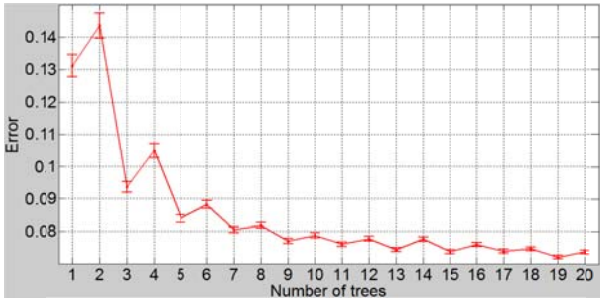
Each experiment results are presented on two graphs. The first one is given for *heart* dataset and the second one shows results obtained on *breast* dataset. It should be noticed that the graphs are examined separately in most of experiments reflecting classifier's behavior on the corresponding dataset. Hence, the axes are not required to be of the same scale.

3.1. Evaluation of RF against the number of trees *m*

The first experiment is aimed to investigate performance of RF against the number of trees *m* from 1 to 20 trees trained and added to the ensemble. The simple voting scheme combines decisions of individual trees. 50 runs of the procedure for cross validation explained above are performed. Test errors are averaged and related confidence intervals are shown on the graphs. Results are presented on *fig. 1*.



a)

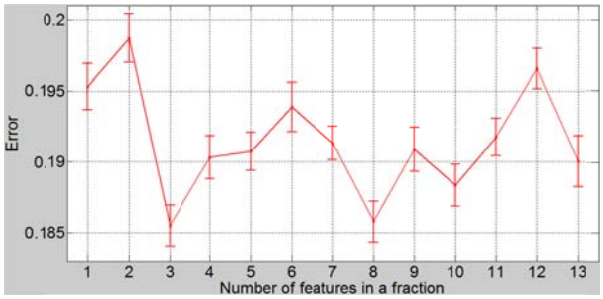


b)

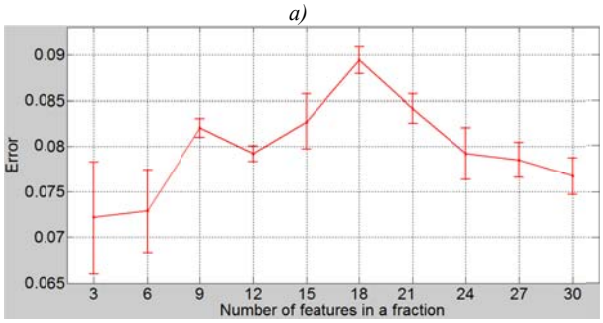
Figure 1. RF test error rate against the number of trees: *a) heart*; *b) breast*

3.2. Evaluation of RF against the number of randomly selected features *k* in a fraction

The second experiment explores performance of RF against the number of randomly selected features *k* in a fraction. The simple voting scheme is used to combine decisions of learners for classification of the examples. Cross validation procedure is executed 25 times. An average RF test error rate and corresponding confidence interval are plotted for each value of *k* from 1 to maximal number of features in the dataset *fnum*, i.e. 13 features for *heart* and 30 features for *breast*. The number of trees *m* in RF is 15. Results are shown on *fig. 2*.



a)

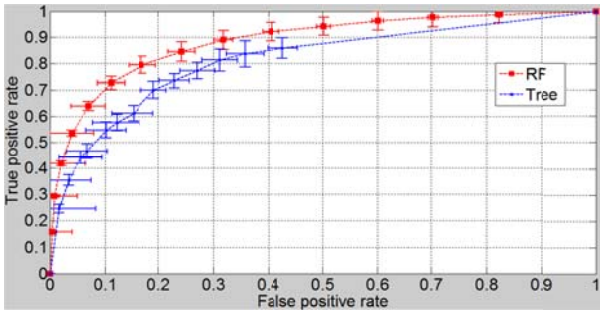


b)

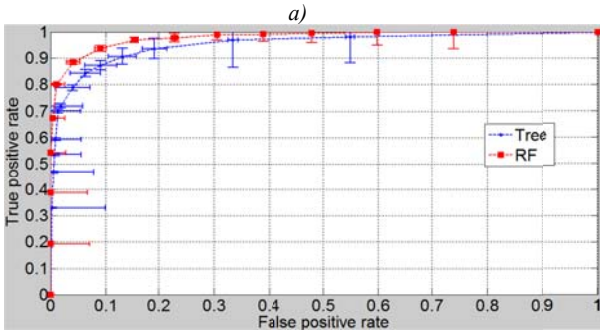
Figure 2. RF test error rate against the number of randomly selected features *k* in a fraction: *a) heart; b)breast*

3.3. Evaluation of RF against a single tree

This experiment is focused on comparison of RF classification performance against a single decision tree. In order to assess classification performance of RF against a single tree, ROC curves with threshold averaging are generated. The number of randomly selected features $k = \sqrt{fnum}$ in a fraction is the same for both RF and the tree, i.e. $k = 3$ for *heart* dataset and $k = 5$ for *breast* dataset. The number of trees m in RF is 15. Bayesian combination scheme is used where prior probabilities are estimated for each tree as train error rates subtracted from 1 and then normalized. The number of two-fold procedure runs is 20. Threshold averaging technique is applied to the set of ROC curves for each of two tested classifiers (i.e. RF and a single tree) and finally obtained ROC curve is plotted on the graph with vertical and horizontal confidence intervals. Results are shown on *fig. 3*.



a)

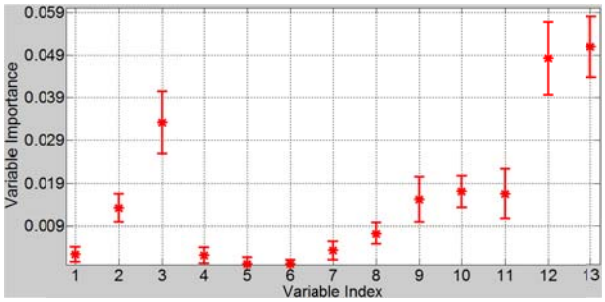


b)

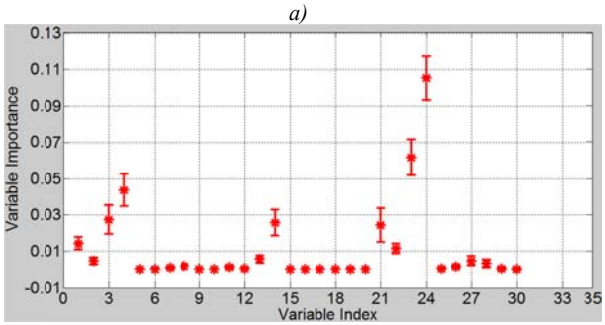
Figure 3. ROC curves with threshold averaging for RF and a single tree: *a) heart; b) breast*

3.4. Estimation of Variable Importance using RF

The aim of this experiment is to measure variable importance on the datasets using RF. The number of trees m in RF is 15. Bayesian combination scheme is used. The approach explained in Background section is implemented on two folds which are generated as described for cross validation. Each time variable importance is estimated on each fold using (8). After 20 repeats of the procedure, averaged variable importance is plotted on the graphs with corresponding confidence intervals. Results are shown on *fig. 4*.



a)

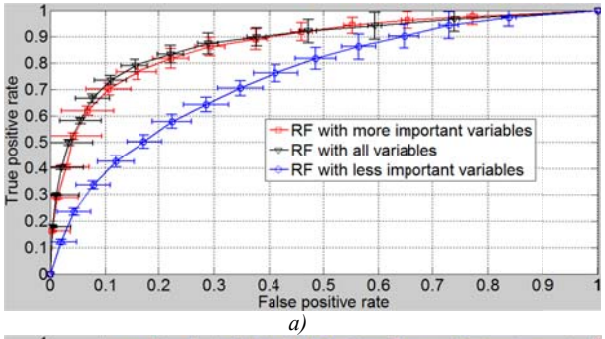


b)

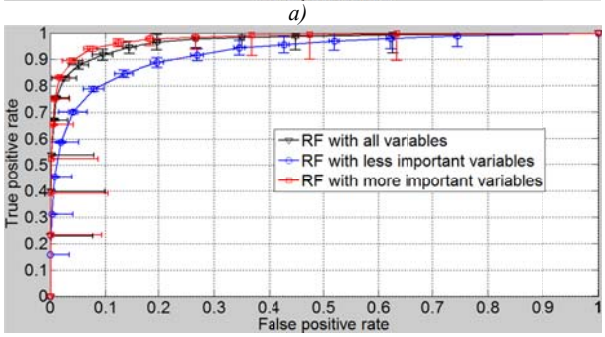
Figure 4. Variable importance using RF: *a) heart; b) breast*

3.5. Variable Selection using RF

Estimation of variable importance using RF allows making conclusions about relative value of a particular set of features in comparison to others from the dataset. This experiment investigates effects and outcomes of variable selection using RF. The number of trees m in RF is 7 for both datasets. Bayesian combination scheme is used. First, RF is trained on the datasets as in previous experiments. The ROC curve with threshold averaging and estimation of variable importance on the dataset using this RF are captured after 20 runs of the procedure on two each time randomly selected folds. After that, the variables are ranked according to the importance rate and divided to two sets: the first one contains 50% of dataset variables with the highest rank (more important variables) while the second one is combined with the rest 50% of variables with lower importance ranks (less important variables). Then the same procedure of training RF with identical parameters, computing the ROC curve, and estimation of variable importance is performed (the number of runs is also the same). The difference is that all steps are carried out for RF trained only using the set of more important variables firstly and the set of less important variables secondly. Thus, three ROC curves are generated in total and presented on each graph. Results of evaluation of RF classification performance for each of three variable selections described above are shown on *fig. 5*.



a)



b)

Figure 5. Variable selection using RF: *a) heart; b) breast*

4. Analysis

Overall, five experiments are performed. Each one is focused on particular points of interest revealing properties or behavior of the RF classifier in particular conditions. Every experiment provides the results of RF evaluation on both *heart* and *breast* datasets. It is reasonable to notice that the classifier's behavior is not always the same on both datasets. The following subsections highlight important points in obtained results and provide explanations for each experiment.

4.1. Evaluation of RF against the number of trees *m*

The experiment results support the statement that the generalization error is decreased as the number of trees *m* in RF is increased. Statistical tests show the same downward trend on both datasets. A one important point to notice is that the average test error has a "zigzag"-like shape with error rises on even numbers of trees and drops on odd numbers. In other words, RF with *m-1* trees performs slightly better than RF with *m* trees if *m* is an even number. This fact can be explained by the nature of voting combination scheme where the final prediction is a majority vote. In other words, if *m* is an odd number, adding a new learner to the ensemble makes it more unstable in its predictions.

4.2. Evaluation of RF against the number of randomly selected features *k* in a fraction

The experiment results show that there is no single trend for RF classification performance while increasing the number of randomly selected features *k* in a fraction. The latter can be explained by the constant tradeoff between increasing the strength and stability of individual tree from one side and consequent growth of the correlation between learners from another, causing the rise of the generalization error upper boundary (4). As the graphs show, the confidence intervals of low values of *k* are wider.

4.3. Evaluation of RF against a single tree

The results of the experiment support the statement that RF has better classification performance in the whole ROC space than a single tree. The ROC curve for RF lays northwest to the tree ROC points on the whole range. Moreover, the ROC curve for RF has considerably more confident intervals for both vertical and horizontal directions.

4.4. Estimation of Variable Importance using RF

The experimental results show the distribution of importance across variables in each dataset. Dataset *heart* has 5 out of 13 variables with comparably low importance rates near 0. In contrast, *breast* is revealed having approximately from 20 to 22 out of 30 variables with comparably little importance just slightly greater than 0. The latter may lead to more efficient variable selection for the second dataset using RF which is the aim of the following experiment to investigate.

4.5. Variable Selection using RF

Estimated variable importance provides capabilities to perform the selection of most important variables in the dataset and the elimination of least important respectively. Therefore, the experiment is focused on exploring effects and outcomes caused by selecting the subset of variables based on the importance rates and how it affects classifier's classification performance.

Overall, the experimental results show that RF which is trained using the subset of less important variables has significantly lower performance rates in ROC space than ones which are trained using all variables or the subset of more important variables as all ROC points are positioned southeast to ones of these two classifiers.

Another noticeable point is that selecting 50% of all variables with higher importance ranks and training RF on this variable subset affects differently its classification performance if it is compared with the one of RF trained

using all variables. Elimination of 50% less important variables causes a slight decline of RF classification performance on *heart* dataset which can be explained by the distribution of importance for this dataset shown in the previous experiment: there are only approximately 5 out of 13 variables with considerably lower importance than others (~ 38%). Therefore, roughly 12% of variables are eliminated regardless with valuable importance. It causes decreasing of classification rates in ROC space.

In contrast, removing 50% of less important variables on *breast* dataset leads to rising of RF performance. The distribution of variable importance in this dataset is approximately characterized by 20-22 out of 30 variables with comparably lower performance (~ 67-73%), so elimination of 50% of less important variables does not result in removing comparably valuable variables. In opposite, it causes ignoring the variables which do not affect the classification rates but may cause the splits during training which might lead to misclassifications while not using valuable variables.

5. Conclusions

The aim of this work is to evaluate RF classification performance and investigate its properties on the given datasets. Experiment 1 and 2 are focused on two main parameters of the RF algorithm: the number of trees m and the number of randomly selected features k on each split. Declining of the average test error with adding new trees is shown on both of datasets. Increasing k results in growth of correlation between learners which is shown by fluctuations of the average error. The lower values of k have more unstable error rates which are shown by wider confidence intervals.

Experiment 3 supports the statement that RF has more favorable classification performance than a single tree classifier that is presented in respective ROC curves.

Experiment 4 shows capabilities to apply RF not only for classification purposes but for estimation of relative variable importance in the given dataset as well.

Experiment 5 reveals how variable selection based on variable importance ranks may affect RF classification performance. Although selection of more important variables does not lead to significant rises of RF classification rates in ROC space, it reduces the number of calculations significantly and may be used for the algorithm optimization while retaining its classification strength that is valuable for practical applications.

To conclude, all properties outlined so far make RF an efficient and powerful classifier with accuracy comparable or even more favorable than one of other state-of-the-art classifiers. Moreover, RF is an easily parallelized and essentially multi-core friendly algorithm since trees can be trained simultaneously on separately generated bootstrap samples. The latter additionally increases its popularity in practical machine learning applications.

References

Breiman, L. Random Forests, *Machine Learning,* vol. 45, pp. 5–32, 2001.

Robnik-Sikonja, M. Improving Random Forests, *ECML Proceedings*, Machine Learning, Springer, Berlin, 2004.

Alpaydin, E. *Introduction to Machine Learning*, MIT Press, Cambridge, Massachusetts, 2nd edition, 2010.

Genuer, R., Poggi J.M., Tuleau-Malot C. Variable Selection Using Random Forests, *Pattern Recognition Letters*, vol. 31, pp. 2225-2236, 2010.

Fawcett, T. An Introduction to ROC Analysis, *Pattern Recognition Letters*, vol. 27, pp. 861-974, 2006.