On the Smart Trees and Competence Areas Based Decision Forest

Vladimir I. Donskoy
V. I. Vernadsky Crimean University, 4, Vernadsky Avenue, Simferopol, 295007, Russia.
vidonskoy@mail.ru, http://www.eng.cfuv.ru/academic-units-and-branches

Abstract: *In this paper, we propose a new approach to decision forest building based on the use of the areas of competence of individual decision trees which are thoroughly trained. We offer two main strategies for the use of a set of decision trees. The first strategy is to use the best decision tree, except the case of his incompetence. The second strategy uses a weighted voting of trees which is organized by a special way.*

***Keywords*:** Decision Forest, Machine Learning, MDL principle.

1. INTRODUCTION

In recent years, to solve classification tasks based on machine learning is widely used decision forests [3]. Decision forest (usually write "random forest" taking into account most often used strategy of the building of decision forest) is decision rule consisting in the use of committee (or ensemble) [11] of decision trees. The random forest building algorithm combines two key ideas: method of bagging and random subspaces method.

Usually all the trees of the random forest are built independently from each other according to the procedure composed of the following three stages:

1^o Generate a random subsample $\mathcal{S}$ of the learning sample with repetitions.

2^o Select a random subset $\mathfrak{F}$ of features.

3^o Build the next decision tree by using selected features $\mathfrak{F}$ and subsample $\mathcal{S}$ (arbitrary splitting criteria [4] may be used but no pruning).

4^o Check the termination condition of the synthesis of the trees and either go to 1^o or terminate the procedure.

Object classification by using random forest can be carried out by (weighted) voting: every tree of the Committee attributes the classified object to one of the classes, and as a result is defined the class, which was elected by the largest number of trees.

Recall that Boosting is a sequential procedure of composition of machine learning algorithms, where each next algorithm seeks to compensate for the shortcomings of the composition of all previous algorithms. Boosting can be viewed as a greedy algorithm which is used to find a good decision composition. Boosting over decision trees which is used for the random forest synthesis is considered to be one of the most effective methods in terms of classification quality. In many experiments it was observed reduction of the error rate on an independent test sample with increase the number of trees included in the composition but not always [2,8].

Describing procedure of boosting, authors often talk about some weak classifiers (or weak learners) used in the committee. The questions naturally arise: why just weak learners? Maybe smart learners are better? Maybe incompetent classifier does not need? Will doctors gather such Concilium in which weak experts will be included? Do not rely on the opinion of the most competent of them?

The idea presented first by L. A. Rastrigin [9] and much later in the paper [7] is that each classier has a particular subdomain (the area of competence) for which it is most reliable. Therefore each classier describes its area of expertise. Given such a description, decision making is realized by using the most reliable classier for the examples in each subdomain. In experiments [7] it was found that such decision to be significantly more effective than various voting techniques which do not seek out subdomains of expertise.

Slightly changing the idea of L.A. Rastrigin, we can say that each classifier should not be applied outside of its area of competence. This leads to the sequential synthesis of classifiers in the following way. At first we build the best classifier and its area of incompetence. From the area of the incompetence of the first classifier specifies a reference to the second classifier, which is built the best from the remaining resources and so on until following classifier can be built.

Concluding the introduction, we note that all above described ideas of building decision forests are heuristic, not having a strict justification.

2. STRONG CLASSIFIERS VS WEAK ONES

We'll use the following widespread notations.

$\boldsymbol{S} = \{(\boldsymbol{x_1}, \boldsymbol{y_1}), \dots, (\boldsymbol{x_l}, \boldsymbol{y_l})\}$ is the learning set consisting of $\boldsymbol{l}$ examples from some fixed, but unknown, distribution $\boldsymbol{\mathcal{D}}$; $\boldsymbol{x_i} \in \boldsymbol{X^n}$, $\boldsymbol{i} = \boldsymbol{1}, \dots, \boldsymbol{l}$; $\boldsymbol{y_i} \in \{-\boldsymbol{1}, +\boldsymbol{1}\}$, $\boldsymbol{y_i} = \boldsymbol{f}(\boldsymbol{x_i})$, where $\boldsymbol{f}: \boldsymbol{X^n} \to \{-\boldsymbol{1}, +\boldsymbol{1}\}$ is the true but unknown classifying function. Thus, $\boldsymbol{X^n}$ is the space of object descriptions, and each object $\boldsymbol{x_i}$ is $\boldsymbol{n}$-dimensional vector from $\boldsymbol{X^n}$; $\{-\boldsymbol{1}, +\boldsymbol{1}\}$ are more convenient label values which define belonging objects to the classes; $\boldsymbol{h_t}: \boldsymbol{X^n} \to \{-\boldsymbol{1}, +\boldsymbol{1}\}$ is a single decision tree, or elementary classifier, $\boldsymbol{t} = \boldsymbol{1}, \dots, \boldsymbol{T}$; $\boldsymbol{\epsilon_t}$ is an error (empirical error) calculated by the sample or part thereof; $\boldsymbol{\mathcal{H}}: \boldsymbol{X^n} \to \{-\boldsymbol{1}, +\boldsymbol{1}\}$ is the final hypothesis which use the set of elementary classifiers, i.e. in our case is the resulting decision forest; $\boldsymbol{p_t} = \boldsymbol{P_{\mathcal{D}}}(\boldsymbol{h_t}(\boldsymbol{x}) \neq \boldsymbol{\mathcal{H}}(\boldsymbol{x}))$ is a probability error of the classifier $\boldsymbol{h_t}$; $\boldsymbol{p_{\mathcal{H}}} = \boldsymbol{P_{\mathcal{D}}}(\boldsymbol{f}(\boldsymbol{x}) \neq \boldsymbol{\mathcal{H}}(\boldsymbol{x}))$ is a probability error of decision forest.

Freund and Schapire [6, Theorem 6] proved that the training error $\boldsymbol{\epsilon_{\mathcal{H}}}$ (the fraction of mistakes on the learning set when AdaBoost is used) of the final hypothesis $\boldsymbol{\mathcal{H}}$ satisfies the inequality

$$\boldsymbol{\epsilon_{\mathcal{H}}} \leq \prod_{\boldsymbol{t}=\boldsymbol{1}}^{\boldsymbol{T}} \left(\boldsymbol{2}\sqrt{\boldsymbol{\epsilon_t}(\boldsymbol{1} - \boldsymbol{\epsilon_t})}\right) \qquad (\boldsymbol{1})$$

where $\boldsymbol{\epsilon_t}$ is an empirical error of the classifier $\boldsymbol{h_t}$. It is obvious that this estimate applies to the boosting of "weak" classifiers which are decision trees.

It is easy to see, that if $\mathbf{0} < \boldsymbol{\epsilon_t} < \mathbf{0,5}$ and the value $\boldsymbol{\epsilon_t}$ will decrease then estimation (1) will also decrease for any given $\boldsymbol{t}$ from the set $\{\mathbf{1}, \dots, \boldsymbol{T}\}$.

Indeed, the function $\boldsymbol{\varphi(\epsilon_t) = \epsilon_t(1 - \epsilon_t)}$ is monotone increasing on the variable $\boldsymbol{\epsilon_t}$ in the interval $(\mathbf{0}, \mathbf{1})$, positive, and less 1 on it. Therefore, a decrease of $\boldsymbol{\epsilon_t}$ on this interval leads to the decrease of $\boldsymbol{\epsilon_{\mathcal{H}}}$. So, why these classifiers $\boldsymbol{h_t}$, $\boldsymbol{t = 1, \dots, T}$, should be weak?

The same result holds if instead of empirical frequencies $\boldsymbol{\epsilon_t}$ consider the probability of relevant events $\boldsymbol{p_t}$. In the paper [10] it is shown that the error probability of the optimal collegial decision of arbitrary independent family of classifiers $\boldsymbol{\mathcal{H} = \{h_t,\ t = 1, \dots, T\}}$ satisfies the inequality

$$\boldsymbol{P_{\mathcal{H}}^{opt} \le \frac{m}{2m-1}\binom{n}{\lfloor n/2 \rfloor}\prod_{t=1}^{T}\sqrt{p_t(1-p_t)}}, \qquad (\mathbf{2})$$

where $\boldsymbol{p_t}$ is a probability of the correct decision by classifier $\boldsymbol{h_t}$, and $\boldsymbol{\frac{1}{2} < m \le p_t < 1}$. The inequality (2) confirms the appropriateness of the selection as accurate as possible independent classifiers in the composition of the decision making committee. These are the classifiers we call strong.

3. DECISION FOREST WITH AREAS OF INCOMPETENCE

When a tree included in decision forest is built, pruning is usually not used. But pruning is a very important element in the procedures of synthesis of decision trees because a branch is pruned when it corresponds to the choice of unreliable decisions.

Pruning of the tree branch is realized when this branch has a length exceeding a certain value. The length of the branch is defined as the number of decision vertices (predicates) in it [5].

We propose instead the decision making in unreliable leaf which the branch ends declare this branch as definition of one of the areas of incompetence of the decision tree.

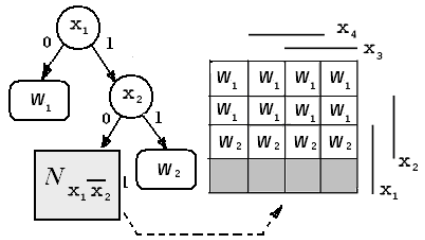

Fig.1 – One simplest example of the area of incompetence

Fig.1 explains how we define of the area of the incompetence of the decision tree with three leaves. The branch $\bar{x}_1$ ends with a leaf w_1 which corresponds to the class with a number 1. The branch x_1x_2 ends with a leaf w_2 which corresponds to the class with a number 2. The branch $x_1\bar{x}_2$ corresponds to the area of incompetence of the tree. In this example, x_1, x_2, x_3, x_4 denote some predicates that are used for splitting.

Area of incompetence appears when the branch of the tree becomes too long and solution which is realized by this branch is ambiguous. Sometimes in the area of incompetence one used a majority decision or determined a vector of empirical frequencies corresponding to conditional probabilities of hitting objects of different classes in this area. We offer to waive the decision in the case of the contact object in the region of incompetence. Instead of continuing of synthesis of the decision tree and growth of branch corresponding to the region of incompetence we propose to start synthesis of the next decision tree – another classifier of the ensemble.

The area of the incompetence of the decision tree can be determined not only by one branch but as well by some of the branches leading to the abandonment decision. The leaves of these branches are marked not by the labels of some classes. Instead the labels pointers to the root node of the next tree are marked. Thus, the area of incompetence is described by a disjunctive normal form over the predicates which are used in the internal nodes of decision trees.

Decision forest with areas of incompetence is shown schematically in Fig.2.

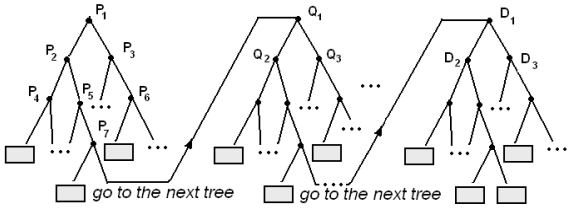

Fig.2 – Decision forest with areas of incompetence

The sets of predicates used to build two different trees of the decision forest (for example $\{P_1, P_2, \dots\}$ and $\{Q_1, Q_2, \dots\}$) may not coincide with each other or coincide partially.

4. WHEN THE BRANCH BUIDING IS ENDED AND A POINTER TO THE NEXT TREE APPEARS?

To answer this question we turn to the MDL (Minimum Description Length) [1] principle. This principle is based on a choice such hypothesis $\hbar^*$ from the hypothesis family $\mathfrak{H}$ which minimizes the sum of the description length of the hypothesis and the code length of the description of learning sample regarding this hypothesis. The corresponding rule of hypothesis choice is determined by the expression

$$\hbar^* = \underset{\hbar \in \mathfrak{H}}{\operatorname{argmin}}(KP(D|\hbar) + KP(\hbar)),$$

where $KP(D|\hbar)$ is a prefix Kolmogorov complexity of the learning sample D regarding to hypothesis $\hbar$, and $KP(\hbar)$ is a prefix Kolmogorov complexity of the hypothesis $\hbar$. However, due to the fact that Kolmogorov complexity is non-computable, instead values $KP(D|\hbar)$ and $KP(\hbar)$ we have to take some their estimators $\widehat{KP}(D|\hbar)$ and $\widehat{KP}(\hbar)$. Recall that MDL corresponds to the Bayesian optimal decision rule, but it avoids the estimation of probability density functions.

In our case, the hypothesis $\hbar$ is a description the current analyzable branch of the tree classifier as the conjunction of some predicates. The magnitude $KP(D|\hbar)$ will be proportional to the number of examples of the learning sample that is incorrectly classified by this branch.

We will need the definition of the algorithmic complexity of a tree branch. To simplify the presentation we assume that all predicates used in internal nodes of the decision trees are of the same type and have the same

complexity $\mathcal{C}_{\mathrm{P}} > 0$. The number of nonterminal nodes in the branch $\mathcal{B}$ increases in the synthesis process of this decision tree. We note this number k_t where t is a step number of the synthesis of the decision tree. Let q_t is a number of examples from the learning sample such these examples are incorrectly classified by this branch $\mathcal{B}$, and $\mathcal{C}_{\mathrm{E}}$>0 is complexity of one example from the learning sample. Total conditional complexity in the region $\Omega_{\mathcal{B}}$ defined by the branch $\mathcal{B}$ is $\mathfrak{C}_t = \mathfrak{C}_t(\mathcal{B})$. The value of $\mathfrak{C}_t$ is determined by the expression

$$\mathfrak{C}_t = \mathcal{C}_{\mathrm{P}} k_t + \mathcal{C}_{\mathrm{E}} q_t \,. \quad (3)$$

If we'll add one more internal conditional node in the branch $\mathcal{B}$ its length will increase. This conditional node splits an area defined by previous state of the branch onto two areas – left and right: Ω_L and Ω_R. Then a number of classification errors will be calculated on two areas by the formula $q_{t+1} = q_{t+1}(\Omega_L) + q_{t+1}(\Omega_L)$.

To decide is it expedient to add one more conditional node we compare complexity of the branch until and after one node is added. The condition of the termination of increasing the length of a branch is determined by the inequality $\mathfrak{C}_{t+1} > \mathfrak{C}_t$ or

$$\mathcal{C}_{\mathrm{P}} k_{t+1} + \mathcal{C}_{\mathrm{E}} q_{t+1} > \mathcal{C}_{\mathrm{P}} k_t + \mathcal{C}_{\mathrm{E}} q_t. \quad (4)$$

Since at each step the branch $\mathcal{B}$ is lengthened on one node i.e. $k_{t+1} - k_t = 1$, inequality (4) takes the form

$$q_t - q_{t+1} < \mathcal{C}_{\mathrm{P}} / \mathcal{C}_{\mathrm{E}} \,. \quad (5)$$

Note that the value $q_t - q_{t+1} = 0$ if the number of errors is not reduced. A value $\Delta q_{t+1} = q_t - q_{t+1}$ is a number on which an error quantity is lessened. Thus, if

$$\Delta q_{t+1} \geq \mathcal{C}_{\mathrm{P}} / \mathcal{C}_{\mathrm{E}} \,. \quad (6)$$

then increasing of the branch $\mathcal{B}$ continues.

As a result we've got the following rule:

1^o if $q_t = 0$ then increasing of the branch is ended and a leaf of this branch is labeled by some class number;

2^o otherwise some predicate is selected for the next splitting and a new node of the tree is added. This conditional node splits an area defined by previous state of the branch onto two areas – left and right: Ω_L and Ω_R. Then an inequality (6) is checked, and either branching is continued or instead a node a pointer to the next tree is placed.

The value $\mathcal{C}_{\mathrm{P}} / \mathcal{C}_{\mathrm{E}}$ is estimated individually for each specific task of machine learning. We propose a formula

$$\mathcal{C}_{\mathrm{P}} / \mathcal{C}_{\mathrm{E}} = \gamma \frac{\mathrm{d} \cdot \log_2 n}{\log_2 l_t}$$

where n is a number of features or dimension of the feature space X^n; d is a number of features used when constructing the splitting predicate; l_t is a length of the part of learning sample which contains only objects which are classified by branch $\mathcal{B}$; γ is a coefficient, which can be tuned to minimize the empirical error by using some optimization procedure.

5. STRATEGIES OF A USE TREES INCLUDED IN THE DECISION FOREST

We offer two main strategies for the use of a set of decisive trees. Both strategies are based on the sequential construction of the trees using the nodal predicates with the best separating capability. It is assumed that the first tree will be built better than the second (will be more competent) and so on.

The first strategy is to use the best decision tree, except the case of his incompetence. Rejection of the solution occurs only in the case when an object enters the areas of the incompetence of all the trees. Otherwise, the class number of the object is determined by the first by order decision tree which is competent for this object.

The second strategy uses a weighted voting of trees. To do this, all the trees that compute for a given object a label of his class are choosing. The weight of the votes for the class with the number J is defined as the sum of the weights of trees, which vote for this class. The heuristic weight $\boldsymbol{W_j}$ of a tree $\boldsymbol{j}$ which outputs a class number of the object $\boldsymbol{x}$ determined by the formula

$$\boldsymbol{W_j = \frac{N_o}{N_j \lambda_j^x}}$$

where $\boldsymbol{N_o}$ is a summarized number of trees included in decision forest; $\boldsymbol{N_j}$ is the number of the tree by the synthesis order; $\boldsymbol{\lambda_j^x}$ is a length of the branch of a tree $\boldsymbol{j}$ which calculates a class number of the object $\boldsymbol{x}$.

6. REFERENCES

[1] A. Barron, J. Rissanen, and B. Yu. The Minimum Description Length Principle in Coding and Modeling, *IEEE Trans. Information Theory* 44 (6) (1998). p. 2743-2760.

[2] S. Bharathidason, C. J. Venkataeswaran. Improving Classification Accuracy based on Random Forest Model with Uncorrelated High Performing Trees, *International Journal of Computer Applications* 101 (13) (2014), p. 26-30.

[3] L. Breiman. Random Forests, *Machine Learning* 45 (1) (2001). p. 5-32.

[4] V. I. Donskoy. Binary decision tree synthesis: splitting criteria and the algorithm LISTBB, *Tavriceskij vestnik informatiki i matematiki* 22 (1) (2013). p. 11-34.

[5] V. I. Donskoy, Yu. Dyulicheva. Decision Forest versus Decision Tree. *Proceedings of the XI-th International Conference "Knowledge-Dialogue-Solution (KDS 2005)"*, Varna, Bulgaria 20-30 June 2005, Vol. 1, pp. 289-297.

[6] Y. Freund, R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *Journal of Computer and System Sciences* 55 (1) (1997), p. 119-139.

[7] J. Ortega, M. Koppel, S. Argamon. Arbitrating among competing classifiers using learned referees, *Knowl. Inf. Syst.* 3(4) (2001). p. 470–490.

[8] T. M. Oshiro, P. S. Perez, J. A. Baranuskas. How Many Trees in a Random Forest? Machine Learning and Data Mining in Pattern Recognition, *Lecture Notes in Computer Science* 7376 (2012), p. 154-168.

[9] L. A. Rastrigin, R. Kh. Erenshtein. Decision processes by a collection of decision rules in pattern-recognition problems, *Automation and Remote Control* 36(9) (1975). p. 1503–1512.

[10] Yu. A. Zuev. On the Estimations of Efficiency of Voting Procedures, *Theory Probab. Appl.* 42 (1) (1997). p. 73-81.

[11] Yu. I. Zhuravlev, S. A. Ablameiko, V.V. Krasnoproshin et all. Algorithms for Algebraic and Logical Correction and their Applications, *Pattern Recognition and Image Analysis* 20 (2) (2010). p. 105-