Proceedings of Machine Learning Research 103:404–413, 2019 ISIPTA 2019

Imprecise Extensions of Random Forests and Random Survival Forests

Lev V. Utkin LEV.UTKIN@GMAIL.COM
Maxim S. Kovalev MAXKOVALEV03@GMAIL.COM
Peter the Great St.Petersburg Polytechnic University (SPbPU), Russia

Anna A. Meldo ANNA.MELDO@YANDEX.RU
St.Petersburg Clinical Research Center for Special Types of Medical Care (Oncological), Russia

Frank P.A. Coolen FRANK.COOLEN@DURHAM.AC.UK
Durham University, United Kingdom

Abstract

Robust weighted aggregation schemes taking into account imprecision of the decision tree estimates in random forests and in random survival forests are proposed in the paper. The first scheme dealing with the random forest improves the classification problem solution. The second scheme dealing with the random survival forest improves the survival analysis task solution. The main idea underlying the proposed modifications is to introduce the tree weights which take simultaneously into account imprecision of estimations as well as aims of the classification and regression problems. The imprecision of the tree estimates is defined by means of imprecise statistical inference models and interval models. Special modifications of loss functions for the classification and regression tasks are proposed in order to simplify minimax and maximin optimization problems for computing optimal weights. Numerical examples illustrate the proposed robust models.

Keywords: classification, survival analysis, random forest, decision tree, deep forest, imprecise Dirichlet model, imprecise probabilities

1. Introduction

The ensemble methodology can be regarded as one of the efficient machine learning approaches to classification and regression. These methods are based on constructing the so-called weak or base classifiers from training data and on aggregating their predictions when classifying unknown samples in order to obtain a strong classifier that outperforms every single one of them. A comprehensive description of ensemble-based models is presented in Zhou's book [35]. One of the best known and most effective ensemble-based models is the random forest (RF) [9], which uses a large number of randomly built individual decision trees in order to combine their predictions. RFs reduce the possible correlation between decision trees by selecting different subsamples of the feature space.

Outputs of decision trees in classification problems are the class probability distributions, which are estimated by the percentage of different classes of examples at the leaf node where the concerned example falls into, and these are combined in order to get the class probability distribution of the corresponding RF. The common combination procedure is the standard averaging of all tree distributions. In order to improve the RF, some weights are assigned to decision trees in accordance with their classification performance. These weights are used in order to replace the standard averaging by weighted averaging of the class probability distributions across all trees. The weights are viewed as additional training parameters.

The main problem of the class probability distributions is that they are assumed to be precise. This is quite restrictive, in particular if there is only a small amount of training data. This makes it interesting to consider the generalization by using imprecise probabilities or imprecise statistical inference models [31].

One of the first ideas of applying imprecise probability theory to decision trees was presented in [2] where probabilities of classes at decision tree leaves are estimated by using an imprecise model, and the so-called Credal Decision Tree model is proposed. Following this work, several papers devoted to applications of imprecise probabilities to decision trees and RFs were presented [3, 4, 25], where the authors developed new splitting criteria taking into account imprecision of training data and the noise data. In particular, the authors consider the application of the Walley's imprecise Dirichlet model (IDM) [32]. The main advantage of the IDM in its application to the classification problems is that it produces a convex set of probability distributions, which has nice properties and depends on a number of observations. Another interesting RF called the fuzzy RF is proposed in [7]. As an alternative to the use of the IDM, nonparametric predictive inference has also been used successfully for imprecise probabilistic inference with decision trees [1]. Imprecise probabilities have been also used in classification problems in [13]. The main focus of interest in this paper is not the decision trees or RFs, but the

weighted averaging procedure which is used to combine the class probability distributions. We study how the use of imprecise distributions may impact on the choice of the corresponding weights of trees.

Another interesting application of RFs is survival analysis [17] when we have censored data, in particular, right-censored data, i.e., times to event of interest for a part of observations or instances are unknown because the events might not have happened during the period of study. Following the well-known Cox proportional hazards model [11], many interesting models using machine learning methods and tools have been proposed (for a comprehensive review see [33]). They include neural networks [14], deep neural networks [27], SVM survival modifications [5], Lasso modification [29], ensemble-based modifications [18].

Due to many advantages of decision trees for classification and regression, several tree-based methods for survival analysis problems have been proposed, for example, [10, 12, 15, 23]. It turned out that a very powerful, efficient and popular tool for survival analysis is the random survival forest (RSF) [8, 19, 20, 21, 26], which can be regarded as a regression model. RSFs require only three tuning parameters to be set (the number of randomly selected predictors, the number of trees grown in the forest, and the splitting rule) [20]. Moreover, RSFs are highly data adaptive and can deal with both low and high dimensional data. The output of a RSF is the cumulative hazard function, which can be computed as the weighted average of the hazard functions obtained as outputs of survival decision trees. Therefore, we also consider how the imprecision of the tree hazard function estimates impact on the weighted procedure of the RSF.

It is important to emphasize that we do not change trees in order to take into account the available imprecision of estimates, but we determine weights in the weighting aggregation procedures to account the imprecision. This is the main idea underlying the proposed robust models. So, we propose robust modifications of RFs and RSFs taking into account imprecision of the decision tree estimates. The important ideas underlying the methods presented in this paper are as follows:

1. Aggregation procedures for computing the RF class probability distributions (classification task) and for computing the RSF cumulative hazard functions (regression task) are modified by introducing the weights of trees.

2. The imprecision of the tree estimates is defined by means of imprecise statistical inference models and interval models, for example, by using the IDM for the classification task and confidence intervals for the regression task of survival analysis.

3. Special modifications of loss functions for the classification and regression tasks are proposed in order to simplify minimax and maximin optimization problems for computing optimal weights.

4. The obtained optimization problems are linear or quadratic with linear constraints.

2. Weighted Averages in Random Forests

In order to consider weighted averaging in random forests, we formally state the standard classification problem. Given N training data (examples, instances, patterns) $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, in which $\mathbf{x}_i$ may belong to an arbitrary set $\mathscr{X}$ and represents a feature vector involving m features and $y_i \in \{1, ..., C\}$ represents the class of the associated examples, the task of classification is to construct an accurate classifier $c : \mathbb{R}^m \to \{1, ..., C\}$ which can predict the unknown class label y of a new observation $\mathbf{x}$ using available training data as accurately as possible, where the accuracy of classification depend on a loss function.

RFs can be regarded as a powerful nonparametric statistical method for both regression and classification problems. Suppose that the RF consists of T trained trees. One of the important peculiarities of decision trees is a probability distribution of classes at each leaf node. This probability distribution is used for computing probabilities of classes for the RF and for making decision about a class label of a testing example. Formally, each leaf node $l \in CL = \{1, ..., L\}$ stores votes for the class labels denoted as $\mathbf{n}_l = (n_{l,1}, ..., n_{l,C})$. Here $n_{l,c}$ is the number of feature vectors from the class c which fall into the l-th leaf node. This is equivalent to storing a categorical probability distribution over classes $c \in \{1, ..., C\}$ in a vector $p_l = (p_{l,1}, ..., p_{l,C}) \in [0,1]^C$. If vector $\mathbf{x}$ falls into the l-th leaf node in a tree, then the prediction of the decision tree for feature vector $\mathbf{x}$ to be of class c is given by $p_{l,c} = n_{l,c}/n_l$. Here n_l is the number of all feature vectors which fall into the l-th leaf node.

One of the ways to improve RFs is to assign weights to decision trees. The weights are used in order to compute the weighted average of the class probability distributions across all trees. They are regarded as training parameters. Their values should minimize the difference between class labels of training examples and values of the RF class distributions. This objective stems from the following reasoning. If an example $\mathbf{x}_i$ has the label $y_i = c$, then we ideally expect that the c-th element of the forest class probability distribution should be close to 1, and other elements of the vector are close to 0. Of course, this condition may be violated. However, this violation could be reduced by controlling the weights for computing the forest class distributions. So, we could find weights of trees in order to minimize the mean difference between class vectors of all examples and the corresponding forest class probability distributions, i.e., the weights can be trained by solving the corresponding optimization problem.

The c-th element $v_{i,c}$ of the class probability distribution produced by the RF for $\mathbf{x}_i$ is determined as

$$v_{i,c} = \frac{1}{T}\sum_{t=1}^{T} p_{i,c}^{(t)}, \quad (1)$$

where $p_{i,c}^{(t)}$ is the probability of class c for $\mathbf{x}_i$ produced by the t-th tree from the RF.

Denote the obtained RF class probability distribution as $\mathbf{v}_i = (v_{i,1}, ..., v_{i,C})$. We propose to change the method for computing $v_{i,c}$, namely, the averaging operator (1) is replaced with the weighted sum with weights $\mathbf{w} = (w_1, ..., w_T)$ of the form:

$$v_{i,c} = \sum_{t=1}^{T} p_{i,c}^{(t)} w_t. \quad (2)$$

Here w_t is a weight of the t-th tree. The weights do not depend on the class c. They are identical for all classes, but different for trees. The weights also do not depend on $\mathbf{x}_i$. They are restricted by the following obvious condition:

$$\sum_{t=1}^{T} w_t = \mathbf{w} \cdot \mathbf{1}^{\mathrm{T}} = 1, \; w_t \geq 0, \; t = 1, ..., T. \quad (3)$$

Here $\mathbf{1}$ is a vector having T unit elements.

3. Training the Weighted Random Forest Classifier

The ideal RF output for $\mathbf{x}_i$, which we denote by $\mathbf{o}_i$ would be when the class probability distribution $\mathbf{v}_i$ is such that it contains a single unit element and $C-1$ zero elements. If the feature vector $\mathbf{x}_i$ belongs to the class y_i, then the class vector is

$$\mathbf{o}_i = (0, ..., 0, 1_{y_i}, 0, ..., 0).$$

Of course, we cannot construct the ideal RF for all training elements, but we can supplement the trees by a weighted averaging procedure that could try to approximate the class probability distribution of the RF to $\mathbf{o}_i$ for every example $\mathbf{x}_i$. We find weights $\mathbf{w}$ such that vectors $\mathbf{v}_i$ will be as close as possible to vectors $\mathbf{o}_i$ whose unit element has the index y_i coinciding with the class label of $\mathbf{x}_i$. This can be done by minimizing the loss function which is defined by the distance $d(\mathbf{v}_i, \mathbf{o}_i)$ between $\mathbf{v}_i$ and $\mathbf{o}_i$, i.e.,

$$J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} d(\mathbf{v}_i, \mathbf{o}_i) + \lambda R(\mathbf{w}).$$

Here $R(\mathbf{w})$ is a regularization term, λ is a hyper-parameter which controls the strength of the regularization. We take the regularization term of the form $R(\mathbf{w}) = \|\mathbf{w}\|^2$ in order to get the quadratic optimization problem.

A simple way is to use the Euclidean distance between two vectors. As a result, we rewrite the loss function for every RF as follows:

$$J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^{N} \sum_{c=1}^{C} \left(\mathbf{p}_{i,c}\mathbf{w} - I_c(y_i)\right)^2 + \lambda \|\mathbf{w}\|^2, \quad (4)$$

subject to (3). Here $I_c(y)$ is the indicator function taking the value 1 if $y = c$, otherwise it is 0; $\mathbf{p}_{i,c} = \left(p_{i,c}^{(1)}, ..., p_{i,c}^{(T)}\right)$. This is a standard convex optimization problem with linear constraints whose solution does not meet any difficulties.

Generally, we can restrict the set of weights by some convex subset $\mathscr{W}(u)$ of the unit simplex in order to improve the regularization. Here u is another regularization parameter which defines the size of the subset $\mathscr{W}$, for example, the hyperparameter of the imprecise Dirichlet model [32] if this model is used for producing the subset $\mathscr{W}(u)$. One of the aims of regularization is to restrict a set of solutions to the optimization problem and to smooth it in order to avoid some outliers. The introduction of the restriction $\mathscr{W}(u)$ for the set of weights plays the same role as the regularization. In sum, the we will assume that constraints to problem (4) are of the form $\mathbf{w} \in \mathscr{W}(u)$.

4. Probabilities of Classes for Trees and Imprecise Probability Models

It is obvious that estimates of class probabilities cannot be considered precise by a small number of training data. Even if we have a lot of training examples, it does not guarantee that many examples fall into a certain leaf node, i.e., n_l is large for all $l \in CL$. This implies that interval-valued or imprecise probabilities $p_{l,c}$ should be taken in place of the precise ones.

Suppose that training example $\mathbf{x}_i$ produces a class probability distribution $\mathbf{P}(i,t) = (p_1(i,t), ..., p_C(i,t))$ at leaf nodes of the t-th decision tree which is unknown precisely, but we know that it belongs to a set $\mathscr{P}_{i,t}(s)$. Here s is a parameter defining the set $\mathscr{P}_{i,t}$. It is assumed that the sets $\mathscr{P}_{i,t}(s)$ are different for different i and t, and they are independent each other. One of the well-known ways for dealing with the imprecise data is to use the minimax or maximin (pessimistic or robust) strategy. In accordance with this strategy, we select a probability distribution from every set of distributions $\mathscr{P}_{i,t}(s)$ such that the loss function $J(\mathbf{w})$ achieves its largest value for fixed values of weights $\mathbf{w}$. It should be noted that the selected "optimal" probability distributions may be different for different values of weights $\mathbf{w}$. In fact, the minimax strategy selects the "worst" distribution providing the largest value of the loss function $J(\mathbf{w})$. Therefore, it can be interpreted as an insurance against the worst case because it aims at minimizing the expected loss in the least favorable case [28]. Robust models have been widely exploited in classification problems due to the opportunity to avoid some strong assumptions underlying the standard classification models [34]. Another "extreme" strategy is optimistic. It selects the "best" distribution providing the smallest value of $J(\mathbf{w})$. It can also be viewed as a direct opposite to the minimax strategy. The optimistic strategy cannot be called robust. Therefore, we do not study it below.

By applying the robust strategy, we write the problem (4) as the maximin optimization problem of the form:

$$J(\mathbf{w}) = \max_{\forall i,t:\mathbf{P}(i,t)\in\mathscr{P}_{i,t}(s)} \min_{\mathbf{w}\in\mathscr{W}(u)} \times \sum_{i=1}^{N}\sum_{c=1}^{C} \left(\mathbf{p}_{i,c}\mathbf{w} - I_c(y_i)\right)^2 + \lambda \left\|\mathbf{w}\right\|^2. \quad (5)$$

Unfortunately, the maximization problem for every i and t is convex. Therefore, its solution can be found on bounds of $\mathscr{P}_{i,t}(s)$. Moreover, attempts to write a dual optimization problem in order to get the minimization problem lead to a non-linear optimization with quadratic constraints. This implies that the direct way for considering the imprecise relaxation of the RF class probability calculation and for computing the optimal weights cannot be applied in a general case except for some simplest cases of the classification problem statement.

Let us diverge from the standard definition of the loss function as the Euclidean distance between vectors $\mathbf{v}_i$ and $\mathbf{o}_i$ and consider an example of the weighted averaging for a RF consisting of $T = 3$ trees and solving two-class classification problem. Suppose that $y_i = 2$. This implies that the vector $\mathbf{o}_i$ is $(0,1)$. Let us suppose that the output of the trees for example $\mathbf{x}_i$ are $p_1 = (0.1, 0.9)$, $p_2 = (0.6, 0.4)$, $p_3 = (0.3, 0.7)$. We would like to make the weighted sum $v_{i,1} = 0.1w_1 + 0.6w_2 + 0.3w_3$ as close to 0 as possible, and the weighted sum $v_{i,2} = 0.9w_1 + 0.4w_2 + 0.7w_3$ as close to 1 as possible. It is important for us to make the second class probability $v_{i,2}$ close to 1. We concentrate only on this objective. In other words, we propose to consider only the weighted sum which corresponds to y_i and should be close to 1. Other weighted sums are not considered. This means that we replace the loss function given above with the following loss function:

$$J(\mathbf{w}) = \min_{\mathbf{w}\in\mathscr{W}(u)} \sum_{i=1}^{N} \left(1 - \sum_{t=1}^{T} p_{y_i}(i,t) w_t\right). \quad (6)$$

One can see from (6) that every term in the objective function contains only the weighted sum corresponding to y_i, i.e., to the unit element of the vector $\mathbf{o}_i$. It should be noted that we do not use the regularization term here because we assume that its function is taken by the subset $\mathscr{W}(u)$. At the same time, a similar problem with the explicit regularization term will be studied later.

4.1. The Linear Programming Problem

The maximin problem can be written now as follows:

$$J(\mathbf{w}) = N - \max_{\forall i,t:\mathbf{P}(i,t)\in\mathscr{P}_{i,t}(s)} \min_{\mathbf{w}\in\mathscr{W}(u)} \sum_{i=1}^{N}\sum_{t=1}^{T} p_{y_i}(i,t) w_t,$$

which is equivalent to

$$J(\mathbf{w}) = \min_{\forall i,t:\mathbf{P}(i,t)\in\mathscr{P}_{i,t}(s)} \max_{\mathbf{w}\in\mathscr{W}(u)} \sum_{t=1}^{T} w_t \sum_{i=1}^{N} p_{y_i}(i,t). \quad (7)$$

Suppose that the set $\mathscr{W}(u)$ is produced by the following linear constraints:

$$a_t \le w_t \le b_t,\ t = 1,...,T,\ \mathbf{w}\cdot\mathbf{1}^{\mathrm{T}} = 1.$$

These constraints correspond to most imprecise statistical models. Let us fix the variables $\mathbf{P}(i,t)$ and write the dual optimization problem for the primal form (7) with $\mathbf{w}$. It is of the form:

$$\min_{f_0,f_t,g_t} \left(f_0 + \sum_{t=1}^{T} (f_t b_t - g_t a_t)\right), \quad (8)$$

subject to $f_t, g_t \ge 0, t = 1,...,T$,

$$f_0 + f_t - g_t \ge \sum_{i=1}^{N} p_{y_i}(i,t),\ t = 1,...,T. \quad (9)$$

Hence, we have two minimization problems (over $\mathbf{P}(i,t)$ and over f_0, f_c, g_c) which can be combined into a single problem taking into account the problem with variables $\mathbf{P}(i,t)$. It is of the form:

$$\min_{\forall i,t:\mathbf{P}(i,t),\ f_0,f_t,g_t} \left(f_0 + \sum_{t=1}^{T} (f_t b_t - g_t a_t)\right), \quad (10)$$

subject to $f_t, g_t \ge 0$, $\mathbf{P}(i,t) \in \mathscr{P}_{i,t}(s)$, and (9).

We have got a linear optimization problem with $NT + 2C + 1$ variables. Since subsets $\mathscr{P}_{i,t}(s)$ are assumed to be different for all $t = 1,...,T$ and $i = 1,...,N$, then we take the smallest value of $p_{y_i}(i,t)$ for every t and i in order to provide the minimum of the objective function. This follows from the fact that the smallest values of $p_{y_i}(i,t)$ make the set of feasible solutions larger. In other words, we can take an extreme point of $\mathscr{P}_{i,t}(s)$ such that $p_{y_i}(i,t)$ is minimal. This implies that we have to find all extreme points of the subset $\mathscr{P}_{i,t}(s)$. It should be noted that several extreme points may have identical largest values of $p_{y_i}(i,t)$. It does not matter because only one probability $p_{y_i}(i,t)$ from all probabilities of the class probability distribution $\mathbf{P}(i,t)$ is used for every i and t.

Let us denote the smallest values of $p_{y_i}(i,t)$ as $p^*_{y_i}(i,t)$. In sum, the smallest probabilities $p^*_{y_i}(i,t)$ have to be found among elements of extreme points. The probabilities do not depend on other variables f_0, f_c, g_c. Therefore, we can return to the primal form (7) and use the smallest values $p^*_{y_i}(i,t)$ in (7). Hence, optimal weights are computed from the linear optimization problem (7) by using the extreme points of $\mathscr{P}_{i,t}(s)$.

4.2. The Quadratic Programming Problem

So far, we have studied how to compute optimal weights of trees under condition that constraints for weights play a role of the regularization. Let us now consider the explicit

regularization $\|\mathbf{w}\|^2$ added to the set $\mathscr{W}(u)$. In this case, we get the following quadratic programming problem:

$$J(\mathbf{w}) = \max_{\forall i,t:\mathbf{P}(i,t)\in\mathscr{P}_{i,t}(s)} \min_{\mathbf{w}\in\mathscr{W}(u)} \times \left(\lambda \|\mathbf{w}\|^2 - \sum_{t=1}^{T} w_t \sum_{i=1}^{N} p_{y_i}(i,t)\right). \quad (11)$$

If we fix the variables $\mathbf{P}(i,t)$, then we have a standard convex quadratic programming problem with linear constraints with respect to $\mathbf{w}$. The problem (11) can be viewed as an extension of (4). Let us again find the dual problem in order to prove that the optimal solution for probability distribution $\mathbf{P}(i,t)$ should be found among largest elements with the index y_i. The dual problem for the minimization problem jointly with the optimization problem over $\mathbf{P}(i,t)$ can be written as

$$\max_{\forall i,t:\mathbf{P}(i,t)\in\mathscr{P}_{i,t}(s)} \max_{\mathbf{v},f_0,f_t,g_t} \left(-\lambda \|\mathbf{v}\|^2 - f_0 - \sum_{t=1}^{T} (f_t b_t - g_t a_t)\right),$$

subject to $f_t, g_t \geq 0$, $t = 1,...,T$, and

$$f_0 + f_t - g_t + 2\lambda v_t \geq \sum_{i=1}^{N} p_{y_i}(i,t).$$

Here $\mathbf{v} = (v_1,...,v_T)$ is a vector of T slack variables, f_t, g_t are non-negative optimization variables, $t = 1,...,T$, f_0 is the optimization variable.

We do not consider the dual form in detail because it is obtained by means of a standard formal procedure. It is important for us to see that the maximum of the objective function is achieved when sums $\sum_{i=1}^{N} p_{y_i}(i,t)$ are minimal. Hence, we substitute the smallest values $p^*_{y_i}(i,t)$ into (11) and solve the following standard quadratic optimization problem for computing optimal wights:

$$J(\mathbf{w}) = \min_{\mathbf{w}\in\mathscr{W}(u)} \left(\lambda \|\mathbf{w}\|^2 - \sum_{t=1}^{T} w_t \sum_{i=1}^{N} p^*_{y_i}(i,t)\right). \quad (12)$$

4.3. A Special Case: The IDM and the Linear-Vacuous Mixture

We consider the IDM [32] and the linear-vacuous mixture or imprecise ε-contaminated models [31, Subsections 2.9.2 and 3.3.5] as special cases of models for defining subsets $\mathscr{P}_{i,t}(s)$ and subset $\mathscr{W}(u)$, respectively. Let us return to the definition of the tree class probability assuming that the i-th training example $(\mathbf{x}_i, y_i)$ falls into a leaf node with number $l(t)$ of the t-th decision tree of a RF. If the vector of stored votes corresponding to this leaf node is $\mathbf{n}_{l(t)} = (n_{l(t),1},...,n_{l(t),C})$, then we can find bounds for probability $p_{l(t),c}$ in accordance with the IDM. It follows from the definition of the IDM that the bounds for the probability are of the form:

$$a_{l(t),c} = \frac{n_{l(t),c}}{n_{l(t)}+s} \leq p_{l(t),c} \leq \frac{n_{l(t),c}+s}{n_{l(t)}+s} = b_{l(t),c}.$$

Here s is the hyperparameter which determines how quickly upper and lower probabilities of events converge as statistical data accumulate [32]. Smaller values of s produce faster convergence and stronger conclusions, whereas large values of s produce more cautious inferences. The detailed discussion concerning the parameter s and the IDM can be found in [6, 32]. In the framework of classification, the hyperparameter s can be regarded as a tuning parameter.

The subset of probability distributions $\mathscr{P}_{i,t}(s)$ has C extreme points $(q_1(t),...,q_C(t))$ such that the c-th extreme point, $c = 1,...,C$, is determined in a simple way as follows:

$$q_k(t) = b_{l(t),c},\ q_c(t) = a_{l(t),c},\ c = 1,...,C,\ c \neq k.$$

It is obvious that the smallest probability $p^*_{y_i}(i,t)$ is equal to $a_{l(t),c}$.

Let us suppose that the set $\mathscr{W}(u)$ is produced by means of the linear-vacuous mixture [31, Subsections 2.9.2 and 3.3.5] with the elicited probability distribution $(T^{-1},...,T^{-1})$ and the parameter $\varepsilon \in [0,1]$, i.e., $u = \varepsilon$. This model can be viewed as another form of the IDM. In particular, there is a connection between parameters s and ε, which is $\varepsilon = s/(T+s)$. It has T extreme points denoted as $\mathscr{E}(\mathscr{W}(\varepsilon))$, which are all of the same form: the k-th element is given by $(1-\varepsilon)T^{-1}+\varepsilon$ and the other $T-1$ elements are equal to $(1-\varepsilon)T^{-1}$, i.e.,

$$q_k(t) = \frac{(1-\varepsilon)}{T} + \varepsilon,\ q_t(t) = \frac{(1-\varepsilon)}{T},\ t = 1,...,T,\ t \neq k.$$

Let us denote

$$A_t = \sum_{i=1}^{N} a_{l(t),y_i} = \sum_{i=1}^{N} \frac{n_{l(t),y_i}}{n_{l(t)}+s}.$$

Then problem (7) can be rewritten by taking into account the extreme points of $\mathscr{W}(\varepsilon)$ as

$$J(\mathbf{w}) = \max_{\mathbf{w}\in\mathscr{E}(\mathscr{W}(\varepsilon))} \sum_{t=1}^{T} w_t A_t = \sum_{t=1}^{T} \frac{(1-\varepsilon)}{T} A_t + \varepsilon \max_{k=1,...,T} A_k.$$

The above implies that the optimal weight vector consists of $T-1$ elements $(1-\varepsilon)T^{-1}$ and one element $(1-\varepsilon)T^{-1}+\varepsilon$. At that, the tree, which provides the largest value of A_t, is assigned by the weight $(1-\varepsilon)T^{-1}+\varepsilon$. This solution is trivial and does not take into account a difference between trees except for one tree with the largest value of A_k. It is given here as an example. The quadratic problem (12) solves this problem and provides better results.

5. Random Survival Forests and Imprecise Models

Let us consider another important problem which deals with survival analysis and is solved by means of RFs. In survival analysis, a patient i is represented by a triplet $(\mathbf{x}_i, \delta_i, D_i)$,

where $\mathbf{x}_i = (x_{i1}, ..., x_{im})$ is the vector of the patient parameters (characteristics) or the vector of features; D_i indicates time to event of the patient, it is assumed to be non- negative and continuous. If the event of interest is observed, D_i corresponds to the time between baseline time and the time of event happening, in this case $\delta_i = 1$, and we have an uncensored observation. If the instance event is not observed and its time to event is greater than the observation time, D_i corresponds to the time between baseline time and end of the observation, and the event indicator is $\delta_i = 0$, and we have a censored observation. Suppose a training set G consists of n triplets $(\mathbf{x}_i, \delta_i, D_i)$, $i = 1, ..., n$. The goal of survival analysis is to estimate the time to the event of interest T for a new patient with feature vector denoted by $\mathbf{x}$ by using the training set G.

The survival function $S(t)$, the hazard function $h(t)$, and the cumulative hazard function $H(t)$ are key concepts in survival analysis for describing the distribution of event times. One of the best models for survival analysis is the RSF due to its properties. A general algorithm for training RSFs is given in detail by Ishwaran et al. [22]. Therefore, we do not consider peculiarities of RSFs and their training.

Let $\{t_{j,k}\}$ be the $N(k)$ distinct death times in terminal node k of the q-th tree such that $t_{1,k} < t_{2,k} < ... < t_{N(k),k}$ and $Z_{j,k}$ and $Y_{j,k}$ equal the number of deaths and patients at risk at time $t_{j,k}$. The cumulative hazard estimate for node k is defined as (the Nelson–Aalen estimator):

$$H_k(t) = \sum_{t_{j,k} \le t} Z_{j,k}/Y_{j,k}. \tag{13}$$

If the i-th patient with features $\mathbf{x}_i$ falls into node k, then we can say that $H(t|\mathbf{x}_i) = H_k(t)$. The ensemble cumulative hazard estimate for the i-th patient (the output of the RSF) is obtained by averaging cumulative hazard estimates of all T trees, i.e.,

$$H(t|\mathbf{x}_i) = \frac{1}{T}\sum_{q=1}^{T} H_q(t|\mathbf{x}_i). \tag{14}$$

The survival function can be obtained from $H(t|\mathbf{x}_i)$ as follows:

$$S(t|\mathbf{x}_i) = \exp\left(-H(t|\mathbf{x}_i)\right). \tag{15}$$

To compare the survival models, the C-index proposed by Harrell et al. [16] is used. The C-index estimates how good the model is at ranking survival times. It estimates the probability that, in a randomly selected pair of patients, the patient that fails first had a worst predicted outcome. In order to define the C-index, we consider admissible pairs $\{(\mathbf{x}_i, \delta_i, D_i), (\mathbf{x}_j, \delta_j, D_j)\}$ for $i \le j$. A pair is not admissible if the events are both right-censored or if the earliest time in the pair is censored. Then the C-index is calculated as the ration of the number of pairs correctly ordered by the model to the total number of admissible pairs. If the C-index is equal to 1, then the corresponding survival model is supposed to be perfect. Let $t_1^*, ..., t_q^*$ denote predefined time points, for example, $t_1, ..., t_N$, where N is distinct event times. If the output of a survival algorithm, for example, a RSF or a survival tree, is the predicted survival function $S(t)$, then the C-index is formally calculated as [33]:

$$C = \frac{1}{M}\sum_{i:\delta_i=1}\sum_{j:t_i<t_j} \mathbf{1}\left[S(t_i^*|\mathbf{x}_i) > S(t_j^*|\mathbf{x}_j)\right]. \tag{16}$$

Here M is the number of all admissible pairs; $\mathbf{1}[a]$ is the indicator function taking the value 1 if a is true, and 0 otherwise; S is the estimated survival function.

5.1. Weights of Survival Decision Trees

In contrast to the classification problem where we assigned weight to minimize the distance between the class distribution and an ideal class vector, in survival analysis, we assign weights to trees in the RSF in order to maximize the C-index. Let us write the C-index as a function of the weights

$$C(\mathbf{w}) = \sum_{i:\delta_i=1}\sum_{j:t_i<t_j} \times \mathbf{1}\left[S(t_i^*, \mathbf{w}|\mathbf{x}_i) - S(t_j^*, \mathbf{w}|\mathbf{x}_j) > 0\right]. \tag{17}$$

Here $S(t_i^*, \mathbf{w}|\mathbf{x}_i)$ is the ensemble survival function obtained by weighted averaging survival function estimates of all T trees taking into account weights $\mathbf{w}$. By maximizing the $C(\mathbf{w})$ over the non-negative weights $\mathbf{w}$ under constraints $\mathbf{w} \in \mathscr{W}(u)$, we can get optimal weights.

First we use (15) to rewrite the C-index through $H(t_j^*, \mathbf{w}|\mathbf{x}_i)$ as

$$C(\mathbf{w}) = \sum_{i:\delta_i=1}\sum_{j:t_i<t_j} \mathbf{1}\left[H(t_j^*, \mathbf{w}|\mathbf{x}_i) - H(t_i^*, \mathbf{w}|\mathbf{x}_j) > 0\right].$$

Let us denote the set of all admissible pairs (i, j) in (17) as J. Then we get the optimization problem:

$$C(\mathbf{w}) = \max_{\mathbf{w}\in\mathscr{W}(u)} \sum_{(i,j)\in J} \times \mathbf{1}\left[\sum_{q=1}^{T} w_q \left(H_q(t_j^*|\mathbf{x}_j) - H_q(t_i^*|\mathbf{x}_i)\right) > 0\right]. \tag{18}$$

In order to solve the problem (18), the indicator function is replaced [30] with the hinge loss function of the form $l(x) = \max(0, x)$. By adding the regularization term, we can write the optimization problem as

$$\min_{\mathbf{w}\in\Delta_T} \sum_{(i,j)\in J} \max\left(0, \sum_{q=1}^{T} w_q \left(H_q(t_i^*|\mathbf{x}_i) - H_q(t_j^*|\mathbf{x}_j)\right)\right) + \lambda \|\mathbf{w}\|^2. \tag{19}$$

5.2. Confidence Intervals for the Nelson–Aalen Estimator

The Nelson–Aalen estimator (13) evaluated at a given time t has a standard $100(1-\alpha)\%$ confidence interval for $H_k(t)$, which is of the form:

$$H_k(t) \pm z_{1-\alpha/2} \cdot \sigma_k(t),$$

where $z_{1-\alpha/2}$ is the $1-\alpha/2$ fractile of the standard normal distribution, $\sigma(t)$ is the variance of the Nelson-Aalen estimator, which is estimated as

$$\sigma_k(t) = \sum_{t_{j,k} \le t} \frac{\left(Y_{j,k} - Z_{j,k}\right) Z_{j,k}}{\left(Y_{j,k} - 1\right) Y_{j,k}^2}.$$

The above implies that $H_k(t)$ has some lower $\underline{H}_k(t)$ and upper $\overline{H}_k(t)$ bounds which have to be taken into account in the problem of computing the weights. By applying the robust strategy, we rewrite the problem (18) as the minimax optimization problem of the form:

$$C(\mathbf{w}) = \min_{H_q(t|\mathbf{x}) \in [\underline{H}_q(t|\mathbf{x}), \overline{H}_q(t|\mathbf{x})]} \max_{\mathbf{w} \in \mathscr{W}(u)} \sum_{(i,j) \in J} \mathbf{1}\left[\sum_{q=1}^{T} w_q \left(H_q(t_j^*|\mathbf{x}_j) - H_q(t_i^*|\mathbf{x}_i)\right) > 0\right]. \quad (20)$$

Unfortunately, the representation (19) cannot be applied because it leads to the quadratic optimization with quadratic constraints. Therefore, we propose to replace the indicator function in (20) with the linear function $l(x) = x$. Note that the C-index increases when the difference of two ensemble cumulative hazard estimates in (20) is positive, and it is not changed when the difference is negative. If we replace the indicator function with the linear one, then we additionally penalize the C-index for the negative difference of the estimates. As a result, we get a more strong condition for the ensemble cumulative hazard estimates. This replacement is not used in classification and regression problems. However, we apply it to the problem of the weight assignment. By using the replacement, we get

$$C(\mathbf{w}) = \max_{H_q(t|\mathbf{x}) \in [\underline{H}_q(t|\mathbf{x}), \overline{H}_q(t|\mathbf{x})]} \min_{\mathbf{w} \in \mathscr{W}(u)} \sum_{q=1}^{T} w_q \sum_{(i,j) \in J} \left(H_q(t_i^*|\mathbf{x}_i) - H_q(t_j^*|\mathbf{x}_j)\right). \quad (21)$$

If we denote

$$B_q = \sum_{(i,j) \in J} \left(H_q(t_i^*|\mathbf{x}_i) - H_q(t_j^*|\mathbf{x}_j)\right) \quad (22)$$

and add the the regularization term $\lambda \|\mathbf{w}\|^2$, then we get the problem similar to (11):

$$C(\mathbf{w}) = \max_{B_q \in \mathscr{B}_q} \min_{\mathbf{w} \in \mathscr{W}(u)} \left(\sum_{q=1}^{T} w_q B_q + \lambda \|\mathbf{w}\|^2\right). \quad (23)$$

Here $\mathscr{B}_q$ is a set of B_q obtained from intervals $[\underline{H}_q(t|\mathbf{x}), \overline{H}_q(t|\mathbf{x})]$. It is obvious that the maximum over B_q is achieved by largest B_q. Therefore, if we find the largest values of B_q, then we get a simple quadratic optimization problem for computing optimal weights, whose solution does not meet any difficulties.

Let us represent B_q as

$$B_q = \sum_{k \in K} b_{k,q} H_q(t_k^*|\mathbf{x}_k).$$

Here $b_{k,q}$ are integers, K is a set of non-identical indices which make up the set J. If $b_{k,q}$ is negative (positive), then we take lower (upper) bound for computing B_q.

6. Numerical Experiments

In order to illustrate the robust classification random forest model, we compare it with the original random forest by using datasets from UCI Machine Learning Repository [24], including Breast Cancer Wisconsin (Diagnostic) ($m = 30$, $N = 569$, $C = 2$), Wholesale Customer Region ($m = 8$, $N = 440$, $C = 3$), Connectionist Bench ($m = 60$, $N = 208$, $C = 2$), Ionosphere ($m = 34$, $N = 351$, $C = 2$). A more detailed information can be found from the data resources. Accuracy measure A used in experiments is the proportion of correctly classified cases on a sample of data. To evaluate the average accuracy, we perform a cross-validation with 100 repetitions, where in each run, we randomly select $N_{\text{tr}} = \gamma N$ training data and $N_{\text{test}} = (1-\gamma)N$ testing data, $\gamma \in [0,1]$ is a parameter which will be used in numerical experiments. We solve the quadratic optimization problem (12) by using the imprecise Dirichlet model with $s = 2$ for $\mathscr{P}_{i.t}(s)$, parameter ε of the set $\mathscr{W}(\varepsilon)$ is selected to get the best accuracy. Different values for the regularization hyper-parameter λ have been tested, choosing those leading to the best results. The RF consists of 100 decision trees. Different values for the RF tuning parameters, including the depth of decision trees, the number of used features for constructing decision trees, have been tested, choosing those leading to the best results.

Numerical results of comparison of classifiers are shown in Figs. 1-3, where the dashed line with triangular markers and the solid line with the circle markers correspond to the original RF and the proposed model, respectively. The figures illustrate accuracy measures of the considered models by different number of training examples, i.e., by different values of γ which changes from 0.05 till 0.95. One can see that the proposed imprecise model outperforms the original RF. The same results are obtained for the Connectionist Bench dataset.

Another interesting question is how the value of s impacts on the accuracy measure. Fig. 4 shows 6 lines corresponding to different s for the Connectionist Bench dataset. This experiment uses $\varepsilon = 0.5$. Moreover, the testing data

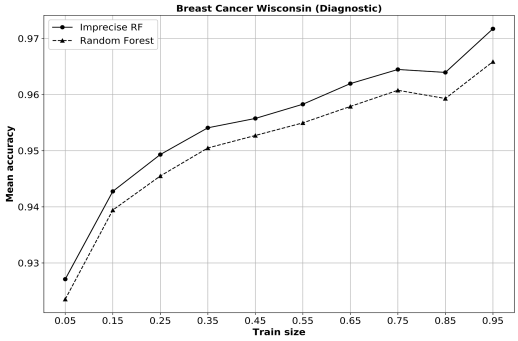

Figure 1: Accuracy measures as a function of γ for the Breast Cancer Wisconsin dataset

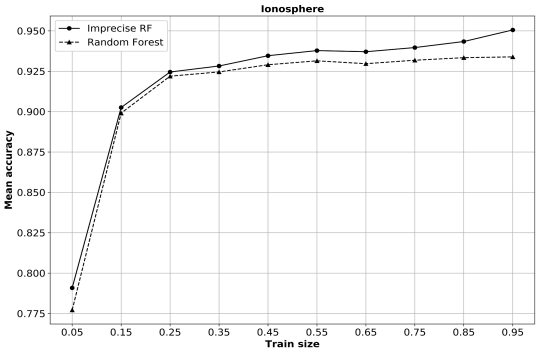

Figure 3: Accuracy measures as a function of γ for the Ionosphere dataset

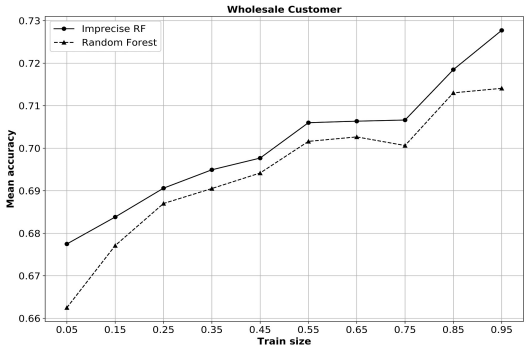

Figure 2: Accuracy measures as a function of γ for the Wholesale Customer Region dataset

are noised such that every feature is randomly changed by using the normal distribution with zero mean and additionally 75% of the feature standard deviation. One can see from Fig. 4 that the accuracy increases with s when the training set is very small. At the same time, large values of s provide worse results by the large amount of training data.

7. Conclusion

An approach taking into account a lack of sufficient data which are required for using precise values of probabilities and hazard functions in ensemble-based models, namely, RFs and RSFs, has been presented. It differs from many available approaches to classification and regression because it does not change trees as week learners, but impacts on their weights which are used for combining the tree outputs. The simple linear and quadratic optimization prob-

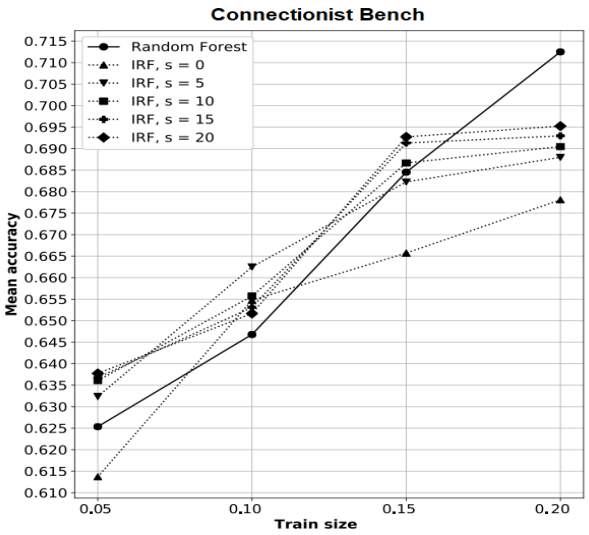

Figure 4: Accuracy measures as a function of s and γ for the Connectionist Bench dataset

lems have been obtained for computing weights optimal to some extent. Only two types of models have been studied: the classification RF model and the regression RSF model. However, the idea underlying the considered models can be simply extended on other ensemble-based models. This is a direction for further research.

By introducing new loss functions, we tried to separate optimization problems in order to simplify the solution and to avoid the useless extremely complex non-linear optimization problems which are unsolvable. Of course, we pay for this simplification by the too pessimistic decision. Perhaps, we would hypothetically get better results by using original loss functions and solving complex non-linear optimization problems. However, our numerical experiments have illustrated that the proposed replacements lead to outperforming results.

It should be noted that we also have applied only two specific loss functions for computing the optimal weights. However, functions different from the given ones can be also applied to simplifying the obtained optimization problems and to getting better results. This is a direction for further research.

We have not provided numerical experiments for the Random Survival Forest due to limited size of the paper. However, they are similar because the corresponding proposed model is based on the same ideas.

Finally, we would like to point out that the idea of introducing the weights for taking into account imprecision can be extended on the functions different from the weighted averaging, for example, on some non-linear functions of a special form or on neural networks which can be also viewed as some complex functions of weights.

Acknowledgement

This work is supported by the Russian Science Foundation under grant 18-11-00078.

References

[1] J. Abellan, R.M. Baker, F.P.A. Coolen, R.J. Crossman, and A.R. Masegosa. Classification with decision trees from a nonparametric predictive inference perspective. *Computational Statistics and Data Analysis*, 71789–802, 2014.

[2] J. Abellan and S. Moral. Building classification trees using the total uncertainty criterion. *International Journal of Intelligent Systems*, 18(12):1215–1225, 2003.

[3] J. Abellan, C.J. Mantas, and J.G. Castellano. A random forest approach using imprecise probabilities. *Knowledge-Based Systems*, 134:72–84, 2017.

[4] J. Abellan, C.J. Mantas, J.G. Castellano, and S. Moral-Garcia. Increasing diversity in random forest learning algorithm via imprecise probabilities. *Expert Systems With Applications*, 97:228–243, 2018.

[5] V. Van Belle, K. Pelckmans, J.A.K. Suykens, and S. Van Huffel. Support vector machines for survival analysis. In *Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare (CIMED2007)*, pages 1–8, 2007.

[6] J.-M. Bernard. An introduction to the imprecise Dirichlet model for multinomial data. *International Journal of Approximate Reasoning*, 39(2-5):123–150, 2005.

[7] P. Bonissone, J.M. Cadenas, M.C. Garrido, and R.A. Diaz-Valladares. A fuzzy random forest. *International Journal of Approximate Reasoning*, 51:729–747, 2010.

[8] I. Bou-Hamad, D. Larocque, and H. Ben-Ameur. A review of survival trees. *Statistics Surveys*, 5:44–71, 2011.

[9] L. Breiman. Random forests. *Machine learning*, 45 (1):5–32, 2001.

[10] A. Ciampi. Generalized regression trees. *Computational Statistics & Data Analysis*, 12:57–78, 1991.

[11] D.R. Cox. Regression models and life-tables. *Journal of the Royal Statistical Society, Series B (Methodological)*, 34(2):187–220, 1972.

[12] R.B. Davis and J.R. Anderson. Exponential survival trees. *Statistics in Medicine*, 8(8):947–961, 1989.

[13] S. Destercke and V. Antoine. Combining imprecise probability masses with maximal coherent subsets: Application to ensemble classification. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*, pages 27–35. Springer, Berlin, Heidelberg, 2013.

[14] D. Faraggi and R. Simon. A neural network model for survival data. *Statistics in medicine*, 14(1):73–82, 1995.

[15] L. Gordon and R.A. Olshen. Tree-structured survival analysis. *Cancer treatment reports*, 69(10):1065–1069, 1985.

[16] F. Harrell, R. Califf, D. Pryor, K. Lee, and R. Rosati. Evaluating the yield of medical tests. *Journal of the American Medical Association*, 247:2543–2546, 1982.

[17] D. Hosmer, S. Lemeshow, and S. May. *Applied Survival Analysis: Regression Modeling of Time to Event Data*. John Wiley & Sons, New Jersey, 2008.

[18] T. Hothorn, P. Bühlmann, S. Dudoit, A. Molinaro, and M.J. van der Laan. Survival ensembles. *Biostatistics*, 7(3):355–373, 2006.

[19] C. Hu and J.A. Steingrimsson. Personalized risk prediction in clinical oncology research: Applications and practical issues using survival trees and random forests. *Journal of Biopharmaceutical Statistics*, 28 (2):333–349, 2018.

[20] H. Ishwaran and U.B. Kogalur. Random survival forests for r. *R News*, 7(2):25–31, 2007.

[21] H. Ishwaran, E.H. Blackstone, C.E. Pothier, and M.S. Lauer. Relative risk forests for exercise heart rate recovery as a predictor of mortality. *Journal of the American Statistical Association*, 99:591–600, 2004.

[22] H. Ishwaran, U.B. Kogalur, E.H. Blackstone, and M.S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2:841–860, 2008.

[23] M. LeBlanc and J. Crowley. Relative risk trees for censored survival data. *Biometrics*, 48(2):411–425, 1992.

[24] M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

[25] C.J. Mantas and J. Abellan. Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data. *Expert Systems with Applications*, 41(5):2514–2525, 2014.

[26] U.B. Mogensen, H. Ishwaran, and T.A. Gerds. Evaluating random forests for survival analysis using prediction error curves. *Journal of Statistical Software*, 50(11):1–23, 2012.

[27] M.Z. Nezhad, N. Sadati, K. Yang, and D. Zhu. A deep active survival analysis approach for precision treatment recommendations: Application of prostate cancer. arXiv:1804.03280v1, April 2018.

[28] C.P. Robert. *The Bayesian Choice*. Springer, New York, 1994.

[29] R. Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[30] L.V. Utkin, A.V. Konstantinov, V.S. Chuknov, M.V. Kots, M.A. Ryabinin, and A.A. Meldo. A weighted random survival forest. arXiv:1901.00213, Jan 2019.

[31] P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.

[32] P. Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 58:3–57, 1996. with discussion.

[33] P. Wang, Y. Li, and C.K. Reddy. Machine learning for survival analysis: A survey. arXiv:1708.04649, August 2017.

[34] H. Xu, C. Caramanis, and S. Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10(7):1485–1510, 2009.

[35] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, Boca Raton, 2012.